

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

Гросуляк Петро Іванович

Система забезпечення наскрізного запису кеш-пам'яті та оперативної пам'яті / The system for the cache memory throughput and RAM ensuring

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

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-42/1
Петро Іванович Гросуляк

Науковий керівник
к.т.н., доцент Масляк Б.О.

ТЕРНОПІЛЬ 2019

РЕЗЮМЕ

Бакалаврська робота містить 63 сторінок пояснюючої записки, 25 рисунків, 8 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломної роботи є розробка системи забезпечення наскрізного запису.

Методи досліджень – методи побудови імітаційних моделей цифрових пристроїв.

В дипломній роботі, на основі аналізу навчальної та наукової літератури, стандартів підготовки фахівців з комп'ютерної інженерії, програм курсу комп'ютерні системи та комп'ютерна схемотехніка, поставлена та реалізована задача розробки системи забезпечення наскрізного запису. В роботі послідовно розглянуті питання методів побудови оперативної пам'яті та кеш-пам'яті. Розглянуто особливості процесора, кеш-пам'яті та оперативної пам'яті. Виділено актуальність розробки системи забезпечення наскрізного запису. Проектування структури контролера системи та деталізація складових створила умови для його реалізації в середовищі NI Multisim.

Ключові слова: НАСКРІЗНИЙ ЗАПИС, ОСНОВНА ПАМ'ЯТЬ, КЕШ-ПАМ'ЯТЬ, МІКРОСХЕМА, СТРУКТУРНА СХЕМА, ЕЛЕКТРИЧНА СХЕМА, КОНТРОЛЕР, КОМБІНАЦІЙНА ЛОГІКА, ДЕШИФРАТОР, РЕГІСТР, ШИННИЙ ФОРМУВАЧ.

RESUME

The diploma project contains 63 pages of explanatory note, 25 figures, 8 tables, 2 application. The volume of graphic material is 2 sheets of A3 format.

The purpose of the thesis is to development of a system for write-through.

Methods of research - methods of constructing simulation models of digital devices.

In the dissertation, on the basis of the analysis of educational and scientific literature, the standards of training specialists in computer engineering, programs of the course computer systems and computer circuitry, the task of developing of a system for write-through. In the work consistently considered the questions of methods of constructing RAM. Features of construction and functioning of interfaces of access to the content of RAM and cache are considered. The topicality of the development of the system for securing the write-up is highlighted. Designing the structure of the system controller and detailing the components created the conditions for its implementation in the medium of NI Multisim.

Key words: WRITE-THROUGH, MAIN MEMORY, CACHE MEMORY, MICROHEMA, STRUCTURE SCHEME, ELECTRICAL SCHEME, CONTROLLER, COMBINATION LOGIC, DECLINE, REGISTER, BUS MOULDER.

ЗМІСТ

Вступ.....	10
1 Особливості застосування та побудови пристроїв оперативної пам'яті	12
1.1 Застосування Кеш-пам'яті в комп'ютерних системах	12
1.2 Методи побудови модулів пам'яті.....	16
1.3 Постановка задачі по реалізації наскрізного запису	22
2 Проектування модуля пам'яті на основі наскрізного запису	24
2.1 Забезпечення когерентності пам'яті на основі наскрізного запису	24
2.2 Технології виводу інформації по шині FSB.....	27
2.3 Структура контролера управління наскрізним записом	30
2.4 Обґрунтування та вибір системи автоматизованого проектування	32
3 Апаратна реалізація методу наскрізного запису	35
3.1 Обґрунтування та налаштування елементів пам'яті в режимі запису інформації.....	35
3.2 Розробка та налаштування контролера управління наскрізним записом...	39
4 Техніко–економічне обґрунтування пристрою	45
4.1 Стадії технологічного процесу	45
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи ...	46
4.3 Розрахунок матеріальних витрат.....	48
4.4 Розрахунок витрат на електроенергію	49
4.5 Обчислення накладних витрат	50

					БР.КСМ. 07110/15.00.00.000 ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Гросуляк П.І.			СИСТЕМА ЗАБЕЗПЕЧЕННЯ НАСКРІЗНОГО ЗАПISУ КЕШ-ПАМ'ЯТІ ТА ОПЕРАТИВНОЇ ПАМ'ЯТІ	Літ.	Арк.	Акрушів
Перевір.		Масляк Б.О.					8	63
Конс.		Паздрій І.Р.				ТНЕУ. ФКІТ. КСМ-42/1		
Н. Контр.		Гураль І.В.						
Затверд.		Березький О.М.						

4.6 Інші витрати.....	51
4.7 Складання кошторису витрат та визначення собівартості.....	51
4.8 Розрахунок ціни проекту	52
4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень	53
Висновки.....	55
Список використаних джерел.....	57
Додаток А Світло-копія публікації	60
Додаток Б Довідка про використання	63

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

Пам'ять є одним із основних вузлів комп'ютера, що призначений для зберігання інформації, тобто програм і даних. Функції пам'яті забезпечуються запам'ятовуючими пристроями, які здійснюють приймання, зберігання і видачу інформації в процесі роботи комп'ютера [1]. Процес приймання інформації в запам'ятовуючій пристрій називають записом, процес видачі інформації – зчитуванням, а спільно їх визначають як процеси звернення до пам'яті.

Звернення процесора до основної пам'яті завжди локалізовано в невеликому діапазоні змін її адрес. Застосування Кеш-пам'яті ґрунтується на обох принципах локальності за зверненням: в ній використовується часова локальність, оскільки зберігається вміст комірок, до яких недавно відбулося звернення, та використовується просторова локальність, оскільки в блоках даних зберігається вміст множини сусідніх комірок [1].

Кеш пам'ять розташована між процесором і основною пам'яттю. Це швидка буферна пам'ять невеликої ємності. Кеш-пам'ять працює на близькій тактовій частоті до процесора і не пригальмовує його роботу. Кеш пам'ять (cache в перекладі з англ. – тайник) залишається прозорою для програміста, тому що система команд процесора, як правило, не містить команд роботи з кеш пам'яттю.

В процесі обчислень процесор не лише зчитує з кеш пам'яті наявну інформацію, але і записує нову, оновлюючи тим самим вміст блоків кеш пам'яті, які є копіями вмісту відповідних блоків основної пам'яті. Виникає ситуація, коли вміст блоку кеш пам'яті та відповідного блоку основної пам'яті перестають співпадати, що створює проблему необхідності фіксування та усунення цього неспівпадіння шляхом оновлення вмісту комірок основної пам'яті. Для подолання цієї проблеми в комп'ютерах з Кеш-пам'яттю використовуються два методи оновлення вмісту блоків основної пам'яті: метод наскрізного запису і метод зворотного запису [1].

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

За методом наскрізного запису перш за все оновлюється вміст комірки основної пам'яті. Якщо в кеш пам'яті існує копія вмісту цієї комірки, то вона також оновлюється [1]. Якщо ж в кеш пам'яті відсутня потрібна копія, то, або з основної пам'яті в кеш пам'ять пересилається вміст блоку, в якому знаходиться оновлене слово (наскрізний запис з відображенням), або цього не робиться (наскрізний запис без відображення).

В ході розробки бакалавратської роботи мною буде опрацьовано матеріал який стосується пам'яті комп'ютера, а зокрема процеси взаємодії між процесором Кеш-пам'яттю та основною пам'яттю. На основі цих даних буде розроблено структурну схему для забезпечення наскрізного запису в кеш та ОЗП. Уже на основі цієї схеми виберуться основні компоненти схеми та зробиться принципова схема для забезпечення наскрізного запису в кеш та основну пам'ять.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ТА ПОБУДОВИ ПРИСТРОЇВ ОПЕРАТИВНОЇ ПАМ'ЯТІ

1.1 Застосування Кеш-пам'яті в комп'ютерних системах

Електронна пам'ять використовується в комп'ютерах та комп'ютерних системах в якості постійної пам'яті, оперативної пам'яті, буферної пам'яті, Кеш-пам'яті та зовнішньої пам'яті [2].

Оперативна, або основна пам'ять (main memory) комп'ютера застосовується для швидкого обміну інформацією (даними та командами) між процесором, зовнішньою пам'яттю (наприклад, дисковою) та периферійними пристроями (комунікації, введення–виведення, графіка) [2]. По іншому її називають – оперативний запам'ятовуючий пристрій (ОЗП), який приблизно співпадає з англійським терміном Random Access Memory (RAM – пам'ять з довільним доступом). Довільність доступу має передбачає можливість операції запису в довільну комірку ОЗП або читання вмісту певної комірки в будь–якому порядку. До основної пам'яті пред'являються такі вимоги:

- великий (як для електронної пам'яті) обсяг, який обчислюється вже десятками, сотнями мегабайт і уже навіть гігабайтами;
- продуктивність і швидкодію реалізації обчислювальної потужності в сучасних процесорах;
- висока надійність зберігання даних – будь–яка помилка навіть в одному з бітів, в принципі, може привести до виникнення помилок в обчисленнях, до спотворення і втрати даних, причому це може статися і на зовнішніх носіях [3].

Кеш-пам'ять (cache memory) – це надоперативная пам'ять, яка слугує буфером між оперативним запам'ятовуючим пристроєм і його «клієнтами» – процесором (навіть декількома процесорами) та іншими абонентами системної шини [4]. Кеш-пам'ять не можна назвати самостійним сховищем, інформація в ньому не може адресуватися клієнтами підсистеми пам'яті, для них присутність кешу є «прозорим». Кеш зберігає в собі копії блоків даних тих областей ОЗП, до

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

яких процесор звертався попередніми разами, оскільки є велика ймовірність подальшого звернення до тих самих даних [4]. Це забезпечує вищу швидкодію забезпечення процесора інформацією з Кеш-пам'яті як з оперативної пам'яті. Велику роль відіграє алгоритм кешування, від ефективності якого залежить вірогідність знаходження в Кеш-пам'яті потрібних даних, а отже, й виграш в продуктивності пам'яті й комп'ютера в цілому. Сучасні процесори мають у своєму розпорядженні вбудовані кеші.

Для енергонезалежного зберігання системної інформації використовується постійна пам'ять – таблиць знакогенераторів, BIOS. Зазвичай при роботі комп'ютера ця пам'ять тільки зчитується, а запис в постійну пам'ять (часто званий програмуванням) часто робиться спеціальними пристроями, які називаються програматорами [4]. Звідси й пішла її назва – Read Only Memory (ROM – пам'ять тільки для зчитування), або ж ПЗУ (постійний запам'ятовуючий пристрій). Обсяг цього типу пам'яті невеликий: наприклад, обсяг пам'яті BIOS PC/XT становив 8 Кбайт, а в сучасних комп'ютерах це значення становить від 128 Кбайт до 2 Мбайт [5]. Швидкодія в постійній пам'яті є нижчою, як оперативній, але це може бути виправлене через застосування тіньової пам'яті. Останніми роками постійну пам'ять витісняє флеш-пам'ять, записування в яку є можливим і в самому комп'ютері через спеціальний режим роботи, та інші типи незалежної пам'яті (FRAM, EEPROM) [5].

Напівпостійна пам'ять переважно використовується щоб зберігати інформацію про конфігурації комп'ютера. Традиційна пам'ять конфігурацій разом із годинником-календарем (CMOS Memory та CMOS RTC) займає обсяг біля кількох десятків байт, ESCD (Extended System Configuration Data) – є областю енергонезалежної пам'яті, яка використовується для конфігурацій пристроїв Plug and Play займає декілька кілобайт. Збереження даних CMOS-пам'яті при відключеному від живлення комп'ютері забезпечується малопотужною внутрішньою батареєю або ж акумулятором [5]. В якості напівпостійної застосовується і електронезалежна пам'ять з довільним доступом

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

(Non-Volatile Random Access Memory, NVRAM), яка зберігає інформацію і при відсутності живлення.

Буферна пам'ять різних адаптерів, а також контролерів (дискових, комунікаційних) здебільшого розподіляється між процесором (а точніше, між абонентами системної шини) і контролерами різних пристроїв [6]. До цього типу пам'яті відносять 16-байтові FIFO-буфери СОМ-порту й кількадекабайтні кеш-буфери високопродуктивних пристроїв зберігання [6]. Специфічним типом буферної пам'яті є відеопам'ять дисплейного адаптера – вона працює в умовах інтенсивного прийому даних з боку ЦП та графічного акселератора одночасно забезпечуючи регенерацію зображення [6].

Основна, або оперативна, пам'ять є одним з так званих «трьох китів», на яких стоїть «комп'ютерний світ» (пам'ять, процесор і периферійні пристрої). Найбільший тягар для оперативного збереження інформації в основному лягає на динамічну пам'ять. Оскільки на сьогоднішній день в неї є найкраще поєднання обсягу, щільності виготовлення, ціни та енергоспоживання. Однак для неї притаманна невисока (за сучасними мірками) швидкодія. Цей недолік можна усунути за рахунок застосування в комп'ютері статичної пам'яті. Статична пам'ять дає комп'ютеру вигреш у швидкодії, але програє динамічній пам'яті у ємності [7].

Швидкодія є одним з найважливіших параметрів пам'яті та визначається часом зчитування даних та виконання операції запису. Складовими цього процесу є тривалість циклу запису/зчитування та мінімальний час доступу до комірки пам'яті [7]. Часом доступу (access time) називають затримку появи потрібних даних на виході пам'яті відносно початку циклу зчитування. Тривалість циклу є мінімальним періодом наступних один за одним звертань до пам'яті, причому можуть бути різні витрати для циклів читання і запису [7]. До циклу звертання окрім активної фази доступу входить ще й фаза відновлення (повернення пам'яті у вихідний стан), яка є співмірною за часом з активною фазою. Часові характеристики запам'ятовуючих елементів визначаються їх технологією виготовлення і принципом дії.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

Продуктивність пам'яті може характеризуватися як швидкість потоку зчитуваних або записуваних даних, що вимірюється в мегабайтах в секунду. Продуктивність підсистеми пам'яті як і продуктивність самого процесора суттєво визначає продуктивність комп'ютера [7]. Виконуючи деякий фрагмент якоїсь програми, процесору доводиться, по-перше, завантажувати з пам'яті відповідний програмний код, а подруге, зробити потрібні обміни даними, і чим менше часу знадобиться підсистемі пам'яті для виконання цих операцій, тим і краще.

Продуктивність пам'яті, як основної, так і кешу, зазвичай характеризують тривалістю пакетних циклів читання (memory burst read cycle). Пакетний режим звернення є основним для процесорів, що використовують кеш (класу 486 і вище); цикли читання виконуються набагато частіше, ніж цикли запису (хоча б тому, що процесору доводиться весь час зчитувати інструкції з пам'яті). Ця тривалість виражається в числі тактів системної шини, необхідних для передачі чергової порції даних в пакеті [8]. Позначення виду 5-3-3-3 для діаграми пакетного циклу читання відповідає п'яти тактам на зчитування першого елемента в циклі і трьом тактам на зчитування кожного з трьох наступних елементів. Перше число характеризує латентність (latency) пам'яті – час очікування даних, наступні – швидкість передачі.

Продуктивність підсистеми пам'яті залежить від таких характеристик як швидкодія і тип застосовуваних запам'ятовуючих елементів, розрядності шини пам'яті і деяких особливостей в архітектурі. Сучасні типи пам'яті мають забезпечувати високу швидкість передачі всередині пакета, використовуючи подвійну або навіть чотирикратну синхронізацію. В цьому випадку параметром шини, по якій передаються дані, може бути як частота передачі даних, так і частота тактового сигналу [8].

В пристроях оперативної пам'яті 2DDR SDRAM або 4 DDR2 SDRAM, системна шина Pentium може в 2 або 4 рази перевищувати тактову частоту. Затримка отримання даних зчитування процесорним ядром в сучасних

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

комп'ютерах може бути від 45 до кількох сотень наносекунд, може змінюватися залежності від способу підключення пам'яті [9].

Застосування різних варіантів конвеєризації підвищує продуктивність мікросхем або модулів пам'яті.

Розрядність шини пам'яті – це кількість байтів (або бітів), з якими операція запису або зчитування може бути виконана за один раз. Розрядність основної пам'яті часто узгоджується з розрядністю сомої зовнішньої шини процесора (1 байт – для 8088; 2 байта – для 8086, 80286, 386SX; 4 байта – для 386DX, 486; 8 байт – для Pentium і вище) [9]. Є очевидним те, що при однаковій швидкодії мікросхем або модулів пам'яті продуктивність блоку з більшою розрядністю є вищою, ніж у меншрозрядних модулів. Саме тому з метою підвищення продуктивності у 32-бітних (по внутрішніх регістрах) процесорів класу Pentium і вище, зовнішня шина, яка пов'язує пам'ять з процесором, має розрядність 64 біта. В сучасних процесорах пропускна здатність шини пам'яті є нижчою ніж пропускна здатність системної шини [9]. Це підштовхнуло розробників до використання двоканальної пам'яті, що є подвоєнням розрядності шини пам'яті щодо розрядності системної шини процесора.

Таким чином, пам'ять є важливим елементом комп'ютерних систем від якого залежить продуктивність роботи.

1.2 Методи побудови модулів пам'яті

Як було показано в попередньому пункті підвищення швидкодії пристроїв пам'яті підвищує швидкість обробки інформації в комп'ютерних системах. Враховуючи, що основна пам'ять комп'ютерів реалізується на повільнішій динамічній пам'яті (DRAM), звертання до неї призводить до певного простою процесора і тому з'являються такти очікування (wait states) [10]. Натомість статична пам'ять (SRAM), як і процесор, побудована, на тригерних схемах та

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

здатна працювати на частотах таких як і в процесора і позбутися тактів очікування (або хоча б скоротити їхню кількість) [10]. Гарним компромісом для побудови економічних і продуктивних систем слугує ієрархічна організація оперативної пам'яті. Поєднання основної пам'яті більш великого обсягу на DRAM з відносно малою Кеш-пам'яттю (КП) на швидкодіючих мікросхемах SRAM є розумним виходом з ситуації [11].

У перекладі слово «cache» (кеш) означає «таємний склад», «зачаток», «тайник». Кеш є додатковим швидкодіючим сховищем копій блоків інформації взятих з основної пам'яті, ймовірність звертання до яких значно вища. Кеш фізично не може в собі зберігати копію всієї основної пам'яті, оскільки його ємність значно менша від ОЗП [11]. Тому він зберігає лише їх обмежену кількість, а також каталог (cache directory) – список поточної відповідності блоків кешу з блоками основної пам'яті.

Більшість сучасних мікропроцесорів для комп'ютерів і серверів мають як мінімум три незалежних кеша: кеш інструкцій для прискорення завантаження машинного коду, кеш даних для прискорення читання і запису даних і буфер асоціативної трансляції (TLB) для прискорення трансляції віртуальних (логічних) адрес в фізичні, як для інструкцій, так і для даних. Кеш даних часто реалізується у вигляді багаторівневого кешу (L1, L2, L3) [12].

Спрощену схему взаємодії КП і ЦП можна описати наступним чином. Спочатку перевіряється наявність потрібної для процесора інформації в найшвидшому кеші – кеші першого рівня, а потім – в кеші другого рівня [13]. Якщо потрібної інформації в цих рівнях кешу не виявиться, то це говорять про помилку, або промах кешу [14]. Якщо інформація в Кеш-пам'яті відсутня взагалі, то процесору доведеться брати її з ОЗП або ж із зовнішньої пам'яті (з жорсткого диска) – рисунок 1.1.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						17
Зм.	Арк.	№ докум.	Підпис	Дата		

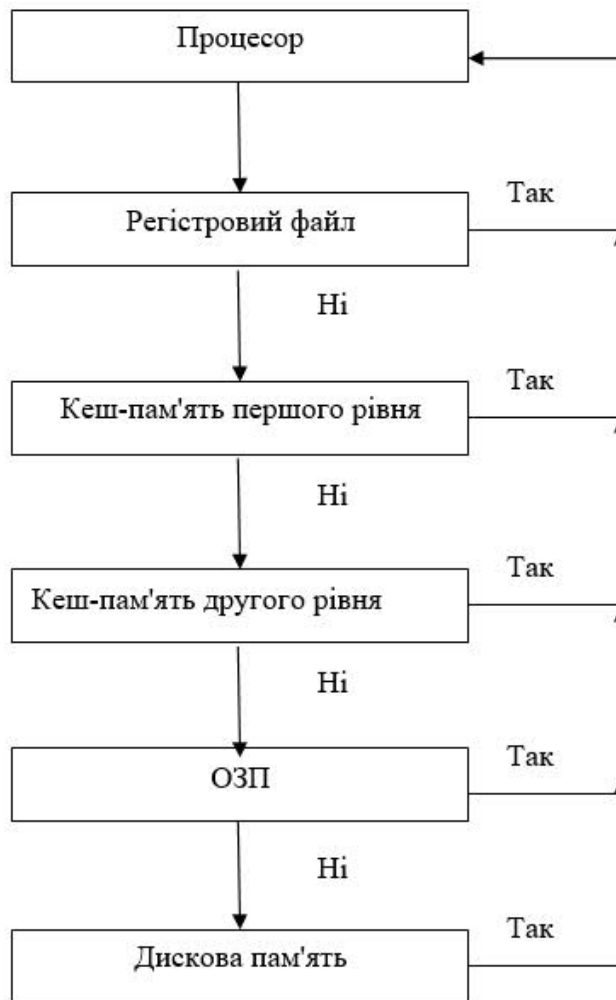


Рисунок 1.1 – Механізм взаємодії процесора з різними рівнями пам'яті

При необхідності заміни інформації в кеші на нову, можуть виконуватися декілька стратегій заміщення. Найбільш відомими серед них є:

- LRU – заміщується рядок, до якого найдовше не звертався процесор;
- FIFO – заміщується найдавніший по перебуванню в кеші рядок;
- Random – заміщення в цій стратегії проходить випадково.

В останньому варіанті, суттєво заощаджуються апаратні засоби в порівнянні з іншими підходами. В ряді випадків метод випадкового заміщення забезпечує більш ефективне використання КП. При кожному із звертань до пам'яті, контролер КП перевіряє чи є копія потрібних даних в кеші. Якщо ж вона там є, то в цей випадок відбувається кеш-попадання (cache hit) і тоді дані беруться з КП [15]. Якщо дійсної копії там нема, то тоді відбувається кеш-

промах (cache miss) і дані тоді будуть братися з основної пам'яті. Відповідно до алгоритму кешування, блок даних, який зчитується з основної пам'яті, заміщає один з блоків Кеш-пам'яті. Від якості алгоритму заміщення залежить відсоток попадань, а отже й ефективність кешування. Існують різні методи заміщення інформації в КП – з прямим відображенням, асоціативний та частково-асоціативний [15]. В КП з прямим відображенням кожен рядок інформації в ОЗП має тільки одне фіксоване місце, на якому можуть перебувати дані в кеш-пам'яті – рисунок 1.2.

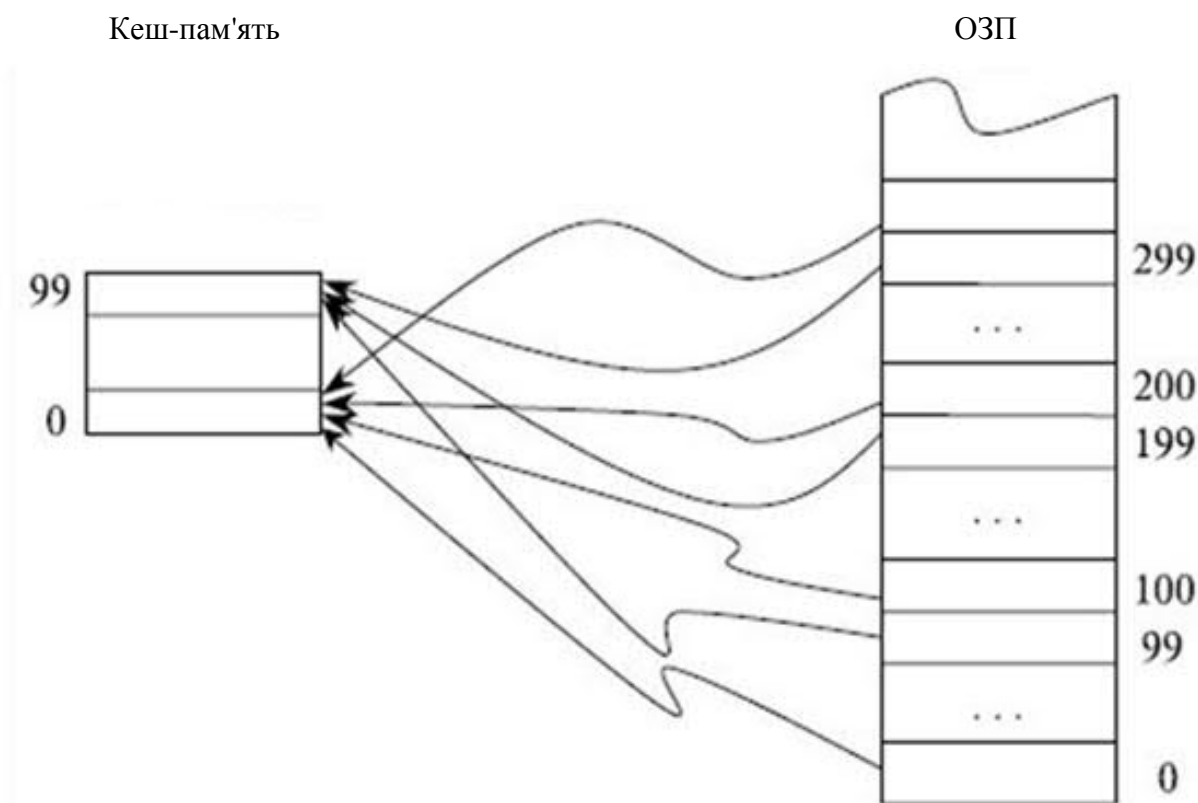


Рисунок 1.2 – Формування вмісту КП

Якщо припустити, що ОЗП складається з 1000 рядків з нумерацією від 0 до 999, а кеш натомість має ємність тільки 100 рядків. У кеші з прямим відображенням рядків ОЗП з номерами 0, 100, 200, ..., 900 можуть зберігатися тільки в рядку 0 КП, тоді рядки 1, 101, 201, ..., 901 ОЗП – в рядку 1 КП, тоді рядки ОЗП з номерами 99, 199, ..., 999 можуть зберігатися тільки в рядку 99 кешу [16].

Ця організація кешу забезпечує швидкий пошук в ній потрібної інформації: тому, що необхідно перевіряти наявність інформації лише в одному місці. Однак ємність Кеш-пам'яті при такій організації використовується не повноцінно: не дивлячись уже на те, що частина КП може бути не заповнена, відбуватиметься витіснення з неї важливої інформації при послідовних зверненнях, наприклад, до рядків 101, 301, 101 ОЗУ [16].

Відповідність між даними в оперативній пам'яті і в Кеш-пам'яті забезпечується внесенням змін до тих області ОЗП, для яких дані в Кеш-пам'яті зазнали змін. Тут виникає проблема, що отримала назву когерентність пам'яті (англ. Memory coherence) [16]. Це властивість комп'ютерних систем, що містять більше одного процесора або ядра і мають доступ до однієї області КП, полягає в тому, що змінена одним ядром / процесором комірка пам'яті приймає нове значення для інших ядер / процесорів [16].

У однопроцесорних системах (строгіше – в одноядерних) роботу з пам'яттю виконує один процесорний вузол. Після запису нового значення в комірку пам'яті, доступну за будь-якою адресою, при читанні даних з тієї ж осередки буде отримано записане значення (навіть при наявності кешування) [17].

У багатопроцесорних (багатоядерних) системах кілька процесорних вузлів працюють одночасно і можуть одночасно (паралельно) звертатися до одній комірці пам'яті (для читання або для запису). Вузли можуть одночасно прочитати значення з однієї комірки пам'яті, можуть зберегти прочитане значення в своїх кешах [17]. Як тільки один з вузлів запише в комірку нове значення, значення, збережені в локальній пам'яті інших вузлів, повинні позначатися як застарілі [17]. Необхідний механізм повідомлення всіх вузлів про те, що значення, збережене в їх кешах, застаріло; такий механізм називається протоколом когерентності (англ. memory coherence protocol). Якщо в системі використовується подібний протокол, то кажуть, що система має «когерентну пам'ять» (англ. Coherent memory) [17].

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

Є два основних способи забезпечення когерентності пам'яті: з наскрізним записом (writethrough) та зі зворотним записом (write-back).

В режимі зчитування ці способи працюють однаково. При записі даних з наскрізним записом оновлення основної пам'яті відбувається разом з оновленням даних в кеші [18]. Це трохи зменшує швидкість системи, оскільки мікропроцесор згодом знову може звернутися за тією ж самою адресою для ще одного оновлення інформації, і тоді минуле пересилання рядка кешу в ОЗП виявиться марним. Але при цьому підході вміст відповідних одиноким рядків ОЗП і Кеш-пам'яті завжди буде ідентичним. Це відіграє велику роль в мультипроцесорних системах із загальною оперативною пам'яттю [18].

Кешування зі зворотним записом змінює рядок ОЗП лише при витісненні рядка з КП, наприклад, якщо потрібно звільнити місце щоб записати новий рядок з ОЗП в уже заповнену Кеш-пам'ять [18]. Операції зворотного запису також ініціюються механізмом підтримки узгодженості Кеш-пам'яті при роботі мультипроцесорної системи із загальною оперативною пам'яттю [18].

Проміжне становище між цими підходами займає спосіб, при якому всі рядки, призначені для передачі з КП в ОЗУ, попередньо накопичуються в деякому буфері [19]. Передача здійснюється або при витісненні рядка, як у випадку кешування зі зворотним записом, або при необхідності узгодження Кеш-пам'яті декількох мікропроцесорів в мультипроцесорній системі, або при заповненні буфера [19]. Така передача проводиться в пакетному режимі, що більш ефективно, ніж передача окремого рядка.

Таким чином, існують різні способи та механізми забезпечення взаємодії КП та ОЗП. Актуальною задачею є вибір механізму підтримання когерентності інформації в КП та ОЗП. Як більш надійний метод пропонується реалізувати механізм, що базується на сквозному записі.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Постановка задачі по реалізації наскрізного запису

Метод наскрізного запису, як було вищесказано, передбачає одночасний запис операнда в комірку з тією ж адресою як кеш, так і основної пам'яті. При цьому процесор не лише зчитує з КП наявну інформацію, але і записує нову, оновлюючи тим самим вміст блоків кеш пам'яті, які є копіями вмісту відповідних блоків основної пам'яті [19]. На рисунку 1.3 приведено структуру взаємодії процесора Кеш-пам'яті та ОЗП за допомогою відповідних інтерфейсів.

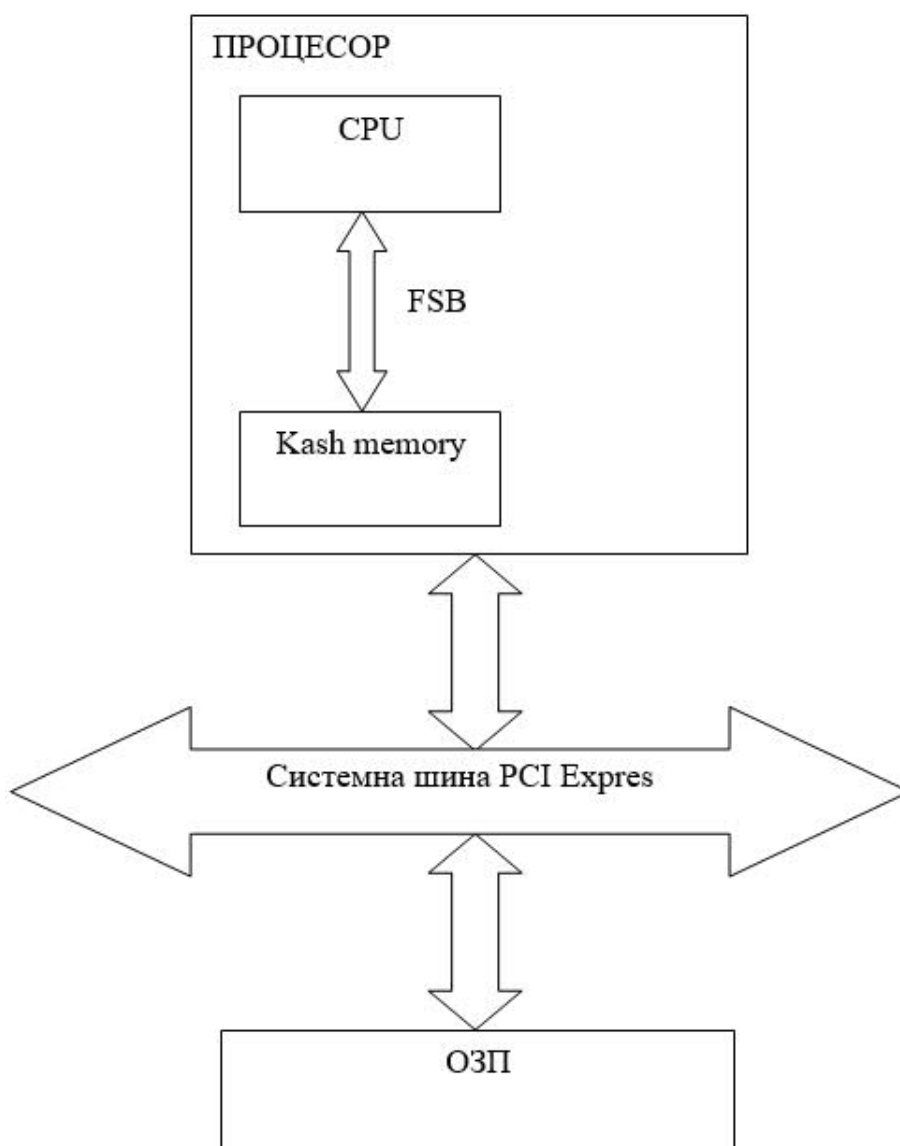


Рисунок 1.3 – Взаємодія процесора та пам'яті через системні інтерфейси

Як видно з приведеної архітектури в наступних розділах слід деталізувати процес взаємодії процесора і основної пам'яті з врахуванням функціонування КП. Для цього слід проаналізувати процес функціонування шини FSB (Front Side Bus).

Дана інформація послужить основою для розробки структури системи забезпечення наскрізного запису (контролера). Розробка структури та її деталізація послужать основою для обґрунтування вибору системи автоматизованого проектування.

В середовищі вибраної системи автоматизованого проектування здійснюється практична реалізація методу наскрізного запису. Для цього слід вибрати та обґрунтувати електричні компоненти, які послужать основою для розробки електричної схеми. Процес розробки та налаштування електричної схеми слід здійснювати шляхом декомпозиції структури на окремі вузли та забезпечити їх синтез.

Таким чином, відповідно до постановки задачі в наступних розділах слід здійснити проектування та практичну реалізацію системи наскрізного запису.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ МОДУЛЯ ПАМ'ЯТІ НА ОСНОВІ НАСКРІЗНОГО ЗАПISУ

2.1 Забезпечення когерентності пам'яті на основі наскрізного запису

За методом наскрізного запису, про який уже згадувалось, передбачається одночасне записування операнда в ту ж комірку пам'яті, що в кеші, і в основній пам'яті [20]. Для організації КП та ОЗП використовують такі основні правила, щоб забезпечити їхню ефективну взаємодію:

- кеш та основна пам'ять діляться на однакові за розміром блоки, тобто вони можуть вміщувати одну і ту ж кількість слів;
- базовою порцією інформації, яка може переміщатися між ОЗП та кешем, є вміст одного блоку;
- кожному блоку присвоєний свій особистий номер;
- комірки пронумеровані в кожному блоці однаково;
- в будь-який момент часу в блоках КП знаходиться скопійований вміст декількох блоків з ОЗП;
- кожне слово в КП супроводжується адресним тегом, яке вказує на те, вміст якого блоку ОЗП є переписаним до блоку КП, в якому знаходиться слово;
- однаковий вміст блоків КП та основної пам'яті забезпечується використанням спеціальних методів оновлення вмісту блоків основної пам'яті;
- заміна вмісту одних блоків в КП вмістом інших блоків з основної пам'яті здійснюється за правилами, які називаються алгоритмом заміщення;
- між блоками КП та основної пам'яті встановлюється відповідність, що задається функцією відображення [20].

Якщо вищеназваних правил дотримано, тоді взаємодія між ЦП, КП та основною пам'яттю відбуватиметься наступним чином. Для початку процесор сформує адреси для взаємодії з основною пам'яттю, не враховуючи наявності кешу. Він видасть адресу комірки основної пам'яті, а за ними сигнали зчитування або запису [20]. Контролер КП визначає, чи дійсно вміст блоку за тою коміркою

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

знаходиться в КП. Якщо знаходиться (тоді відбувається попадання), то при зчитуванні процесор взаємодіє лише з КП, а при записі, щоб зберегти ідентичність вмісту блоків основної пам'яті і кешу може бути два варіанти [20]. При першому варіанті буде здійснюватися одночасний запис операнда в комірку з тією ж адресою в КП та в основній пам'яті – рисунок 2.1.

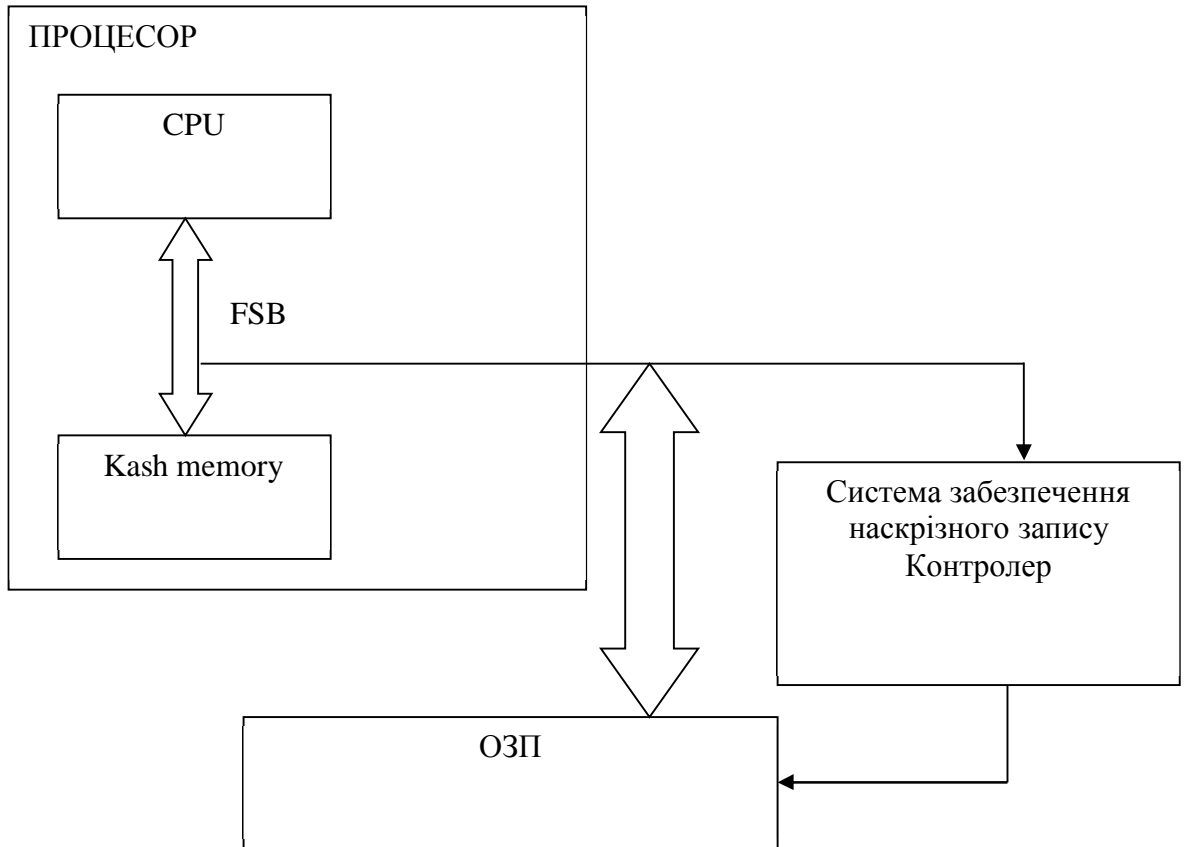


Рисунок 2.1 – Узагальнена структура процесу забезпечення наскрізного запису

Як видно з рисунку 2.1, система забезпечення наскрізного запису повинна перехопити сигнал запису інформації по шині FSB та транслювати його на шину PCI. Оскільки адреси комірок в КП та ОЗП співпадають, то здійсниться одночасне заміщення інформації [20].

При другому варіанті замість одночасного запису буде використовуватися біт модифікації. А саме заміщення вмісту цієї комірки основної пам'яті відбувається дещо пізніше, при заміщенні в КП вмісту цього блоку.

Таким чином, усі сигнали читання/запису, що процесор посилає до основної пам'яті перехоплює КП, а вразі потреби надає процесору копії цих даних, які тимчасово зберігалися у його власній робочій пам'яті [21].

Якщо КП здатна собою підмінити основну пам'ять (це відбувається у понад 96–98 відсотків випадків), тоді вона своїми власними ресурсами задовольняє запит процесора. В цьому випадку процесор не пригальмовує і продовжує працювати на повній швидкості. Коли "підміна" основної пам'яті неможлива (а це менше двох–чотирьох відсотків), тоді КП підключає до роботи основну пам'ять, обмін даними з якою сильно пригальмовує процесор [21]. Усіма перехопленнями запитів від процесора до ОЗП, керується контролером КП, який є його складовою частиною [21]. Ще одною частиною КП є його невелика робоча пам'ять, в якій зберігаються копії вмісту блоків основної пам'яті, які обслуговували останні запити процесора. Важливо, що вміст комірок основної пам'яті копіюється до КП разом із їхніми адресами. Саме ці адреси і дозволяють контролеру КП приймати рішення про спроможність задовольнити конкретний процесорний запит без залучення до обміну повільної оперативної пам'яті [21].

Забезпечення ідентичності вмісту блоків КП і основної пам'яті використовуються метод, який передбачає одночасний запис операнда в комірку як кеш, так і основної пам'яті. Даний метод застосований в мікропроцесорах i486 фірми Intel [22]. Певний вигравш дає модифікація методу наскрізного запису, відома як метод буферизованого наскрізного запису. Інформація спочатку записується в кеш пам'ять і в спеціальний буфер, що працює за схемою FIFO. Запис в основну пам'ять проводиться вже з буфера, а процесор, не чекаючи її закінчення, може відразу ж продовжувати свою роботу [22]. Звичайно, відповідна логіка керування повинна своєчасно очищувати заповнений буфер. При використанні буферизації процесор повністю звільняється від роботи з основною пам'яттю [22].

Таким чином, в процесі реалізації наскрізного запису буде використано системну шину FSB. Завданням наступного розділу буде детальний аналіз

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

процесів виводу інформації по даним шині, який дасть можливість деталізації структури системи забезпечення наскрізного запису.

2.2 Технології виводу інформації по шині FSB

Системною шиною для обміну інформацією між процесором кеш– та оперативною пам'яттю є шина FSB. Аббревіатура FSB розшифровується як Front Side Bus, що можна перевести як «передня» шина. В основні функції шини входить передача даних між процесором і чіпсетом. Точніше кажучи, FSB розташовується між процесором і мікросхемою «північного мосту» материнської плати, де знаходиться контролер оперативної пам'яті [22].

Зв'язок же між північним мостом і іншої важливої мікросхемою чіпсета, званої «південним мостом» і містить контролери пристроїв введення–виведення, в сучасних комп'ютерах зазвичай здійснюється за допомогою іншої шини, яка носить назву Direct Media Interface.

Як правило, процесор і шина мають одну і ту ж базову частоту, яка називається опорною або реальною. У разі процесора його кінцева частота визначається твором опорної частоти на певний множник. Взагалі кажучи, реальна частота FSB зазвичай є основною частотою материнської плати, за допомогою якої визначаються робочі частоти всіх інших пристроїв [22].

У більшості комп'ютерів реальна частота системної шини визначала і частоту оперативної пам'яті, однак зараз пам'ять часто може мати й іншу частоту – в тому випадку, якщо контролер пам'яті розташовується в самому процесорі. Крім того, слід мати на увазі, що реальна частота шини не еквівалентна її ефективній частоті, яка визначається кількістю переданих біт інформації в секунду [2].

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

В даний час дана шина вважається застарілою і поступово замінюється новими – QuickPath і HyperTransport. Системна шина QuickPath є розробкою фірми Intel, а HyperTransport – компанії AMD.

Системи SMP на основі P-III використовують FSB як спільну шину для зв'язку. FSB є 7-ступінчастою конвеєрною шиною, що складається з request1, request2, error1, error2, snoop, response, і фаз даних, як показано на рисунку 2.5. Фази відповіді і даних часто перекриваються. FSB підтримує вісім невиконаних угод. Два активнотривіжні сигнали шини (HIT# і HITM#) призначені для призначення Snooping (Вистежування), це зображено на рисунку 2.5 [23]. Результат Snoop кожної транзакції пам'яті приводиться в фазу обмотки конвеєра.

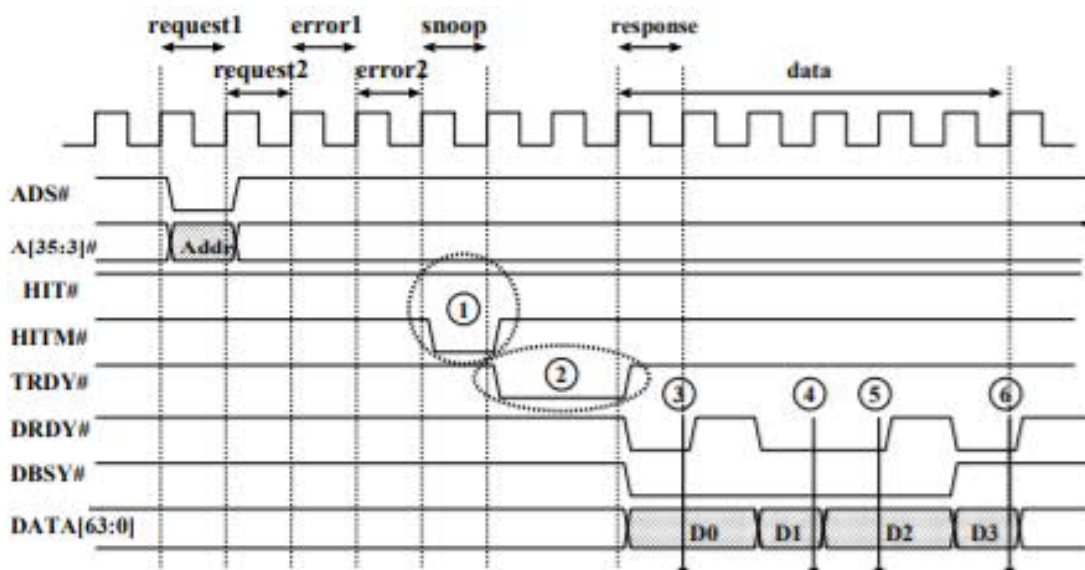


Рисунок 2.5 – Часова діаграма передачі даних на шиною FSB

Щоб оновити пам'ять, контролер пам'яті (MC) повинен бути готовий прийняти дані. Його готовність показує підтверджуючий сигнал #TRDY (цільовий готовий) шини FSB, як зображено на рисунку 2.5. Після цього вісім слів (32В) передаються, як показано в 3, 4, 5 і 6 [23]. Кількість прийнятих шинних циклів залежить від готовності даних від процесора або основної пам'яті. На рисунку 2.5 показано шість циклів для передачі однієї лінії кешу.

Сигнали DRDY # і DBSY # використовуються для інформування часу для фіксації даних P-III та / або MC. На рисунку 2.5 показано використання цих сигналів. Сигнал DBSY # зберігається, поки всі 4 чотирирівневі слова не будуть передані. Потім, коли підтверджується сигнал DRDY #, дані доступні, як показано на рисунку 2.5, останні дані доступні, коли сигнал DRDY # підтверджується і сигнал DBSY # відхиляється [23].

На рисунку 2.6 показаний спосіб записування даних по шині FSB за допомогою наскрізного запису впродовж трьох основних етапів. На 1 етапі процесор звертається до контролера пам'яті для підтвердження запиту на оновлення даних в кеші та ОЗП. В 2 та 3 етапах дані по системній шині записуються спочатку в кеш а потім в самий ОЗП.

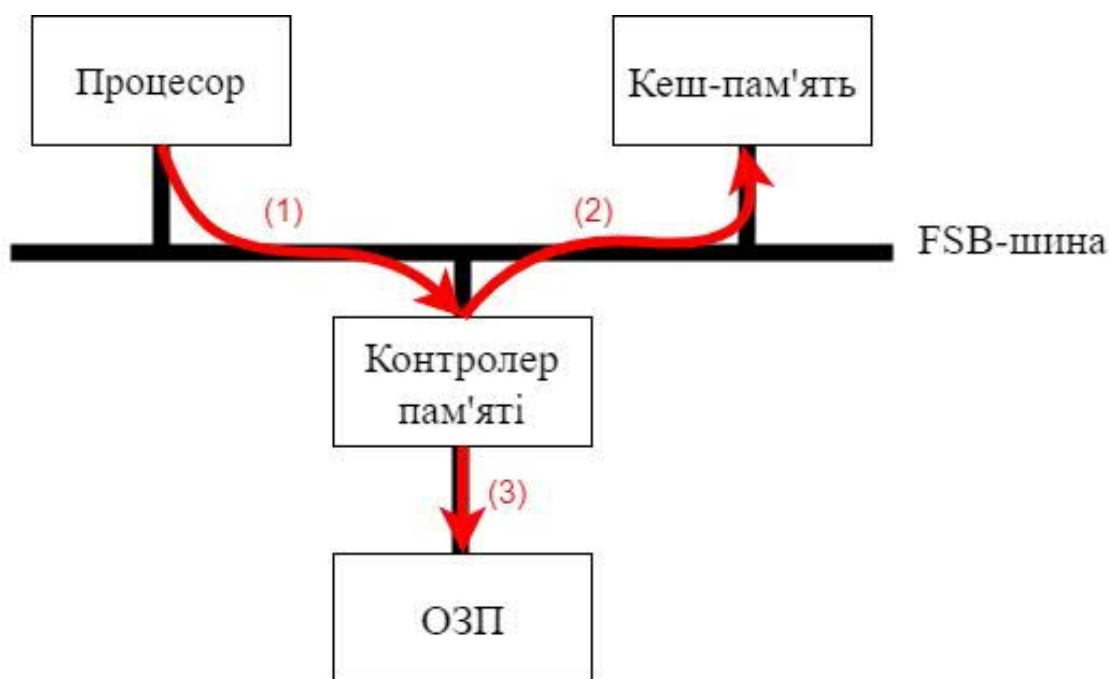


Рисунок 2.6 – Узагальнена структурна схема реалізації наскрізного запису

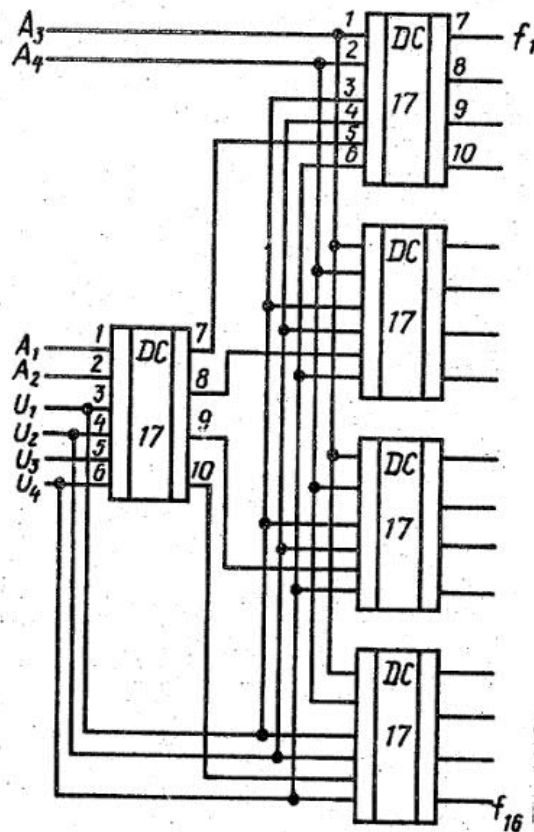
Таким чином, в процесі реалізації наскрізного запису вибранана нами системна шина FSB буде використовуватися для передачі інформації між пристроями. В наступному пункті буде розроблено принцип взаємодії ключових елементів майбутньої схеми, а також загальну структуру системи забезпечення наскрізного запису.

2.3 Структура контролера управління наскрізним записом

Для розробки контролера управління наскрізним записом знадобляться такі елементи як: дешифратор адрес, тригер адрес, регістри, логічне І.

В свою дешифратором для дешифрування адрес буде слугувати пірамідальний дешифратор. Його особливістю є двократне збільшення числа виходів на кожному ступені дешифрування і застосування у всіх щаблях тільки 2-х вхідних ЛЕ [].

Розглянемо принцип побудови 5-ти розрядного пірамідального дешифратора на $N=2^5=32$ виходи. На першій ступені декодування використовується найпростіший повний дешифратор DC 2x4. На кожному з наступних ступенів вводять по одній змінній (з інверсією і без). Число ступенів в пірамідальних дешифраторів на 1 менше розрядності дешифрованого коду [24].
Схема пірамідального дешифратора приведена на рисунку 2.7.



Зм.	Арк.	№ докум.	Підпис	Дата

Рисунок 2.7 – Пірамідальний дешифратор

Ці дешифратори реалізовані в ряді серій ІМС, наприклад: К155ІД1 (дешифратор 4x10), К555ІД6 (дешифратор 4x10), К555ІД3 (дешифратор 4x16), 531ІД14 (подвійний, високошвидкісний дешифратор 2x4), 531ІД4 (дешифратор 2x4) 531ІД10 (дешифратор 4x10) [25].

Функціональна схема схемотехнічної реалізації контролера виводу інформації приведена на рисунку 2.8.

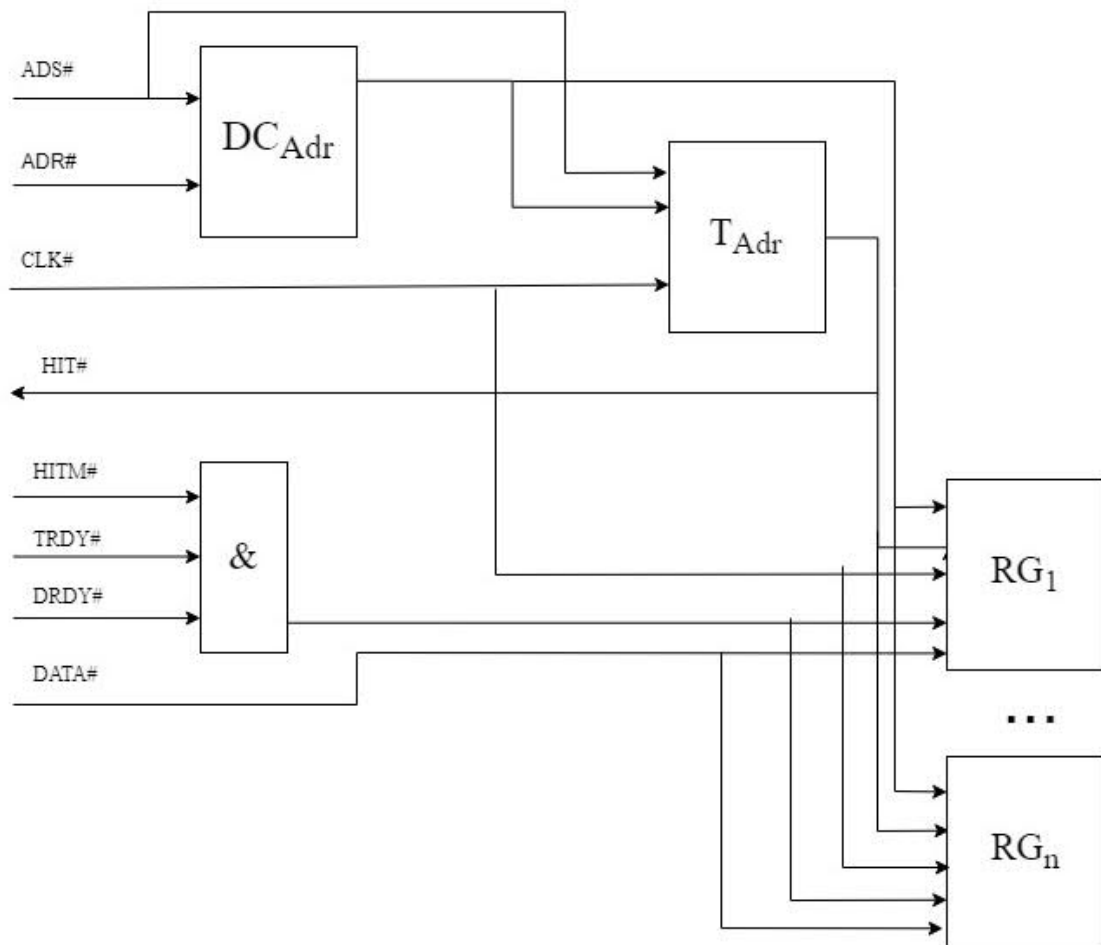


Рисунок 2.8 – Спроектвана структурна схема пристрою управління наскрізним записом

Дані адреси (ADR) разом з дозволяючим сигналом адрес (ADS) поступають на відповідний дешифратор адреси. На виході дешифратора отримується сигнал, який прямує на на вхуд тригера адрес. До входу тригера також приєднуються сигнали ADS та генератор іспульсів (CLK). Після цього на

виході отримується сигнал НІТ, що означає попадання і те що необхідна адреса в регістрах присутня. Цей сигнал відправляється до процесора, щоб повідомити йому про це.

На входи ж регістрів подаються наступні сигнали: вихідні дані дешифрованої адреси з дешифратора, вихідні дані з тригера, генератор імпульсів, логічне І (результат виконання сигналів: НІТМ, TRDY, DRDY) та дані (DATA). Сигнал НІТМ вказує на те що процесор хоче записати дані, а TRDY та DRDY про готовність регістрів їх сприйняти. Коли ці сигнали готові, дешифратор знаходить адресу і тоді почнуть записуватися дані в регістри.

Таким чином, в даному розділі на основі аналізу функціонування системної шини FSB запропоновано структуру пристрою управління наскрізним записом як в кеш–, так і оперативну пам'ять.

2.4 Обґрунтування та вибір системи автоматизованого проектування

Для проектування потрібної схеми було вибрано NI Multisim, що є популярним програмним пакетом, що дозволяє моделювати електронні схеми і друковані плати. Головним чинником вибору цього програмного засобу стало те, що він є уже знайомим для мене засобом моделювання схем.

Головна особливість NI Multisim – простий наочний інтерфейс, потужні засоби графічного аналізу результатів моделювання, наявність віртуальних вимірювальних приладів, які копіюють реальні аналоги. Бібліотека елементів містить понад 2000 SPICE–моделей компонентів National Semiconductor, Analog Devices, Phillips, NXP і інших виробників. Присутні електромеханічні моделі, імпульсні джерела живлення, перетворювачі потужності [26]. Інструмент Convergence Assistant автоматично виправляє параметри SPICE, коригуючи помилки моделювання. NI Multisim випускається в двох варіантах – Professional і Education.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

Версія Multisim Education призначена для навчальних закладів і включає в себе навчальні курси, підготовлені апаратні рішення і робочі підручники. Основне завдання – закріпити теоретичний матеріал, наочно продемонструвавши роботу тих чи інших законів і процесів в реальних проектах [26]. Для цього крім інтерактивних компонентів програма здатна взаємодіяти з апаратними платформами NI myDAQ (бібліотека контрольно–вимірювального обладнання) і NI ELVIS (віртуальний інструментарій для навчальної майстерні), що робить можливим створення цілих віртуальних лабораторії систем управління, енергетики, мехатроніки і силової техніки. На рисунку 2.9 показано робоче вікно програми.

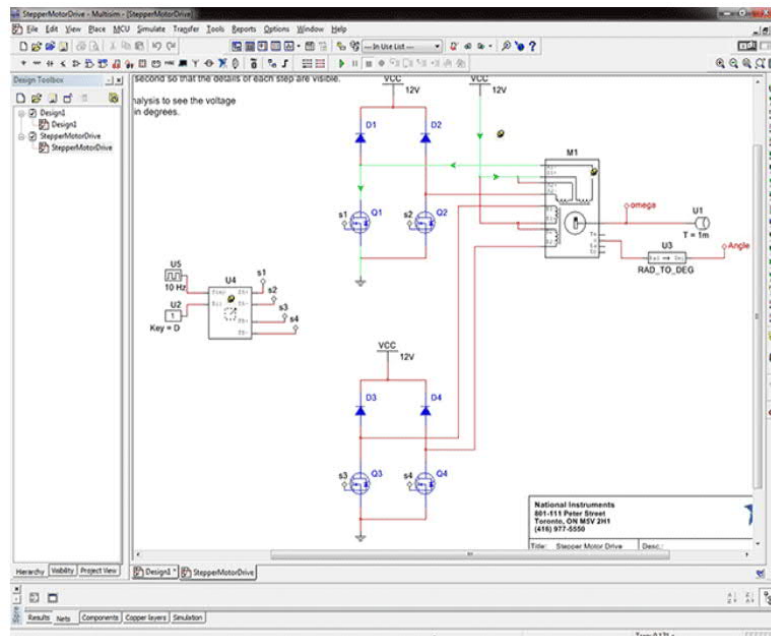


Рисунок 2.9 – Робоче вікно програми NI Multisim

Версія Multisim Professional спеціально створена для швидкого прототипування і рішення задач оптимізації з'єднань. Пропонується розширений призначений для користувача інтерфейс, нестандартні методи аналізу, засновані на фірмовій системі NI LabVIEW, і звичайні алгоритми імітаційного моделювання схем за стандартом SPICE [26].

Останні версії програми мають поліпшеною функціональністю, новими інструментами для моделювання, розширеної базою елементів, завдяки чому

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

розробка і створення проектів електричних схем може виконуватися набагато більш точно і швидко. NI Multisim може взаємодіяти з середовищем розробки систем вимірювання LabVIEW, що дозволяє зіставляти теоретичні дані з реальними, прямо в ході створення схем друкованих плат. Це зменшує кількість проектних помилок і прискорює реалізацію проектів [27]. Зворотною стороною цього стали завищені системні вимоги, що пред'являються до обладнання. Навантаження на процесор і пам'ять при роботі з великими схемами і при трасуванні дуже велика. На рисунку 2.10 показано налаштування компонентів.

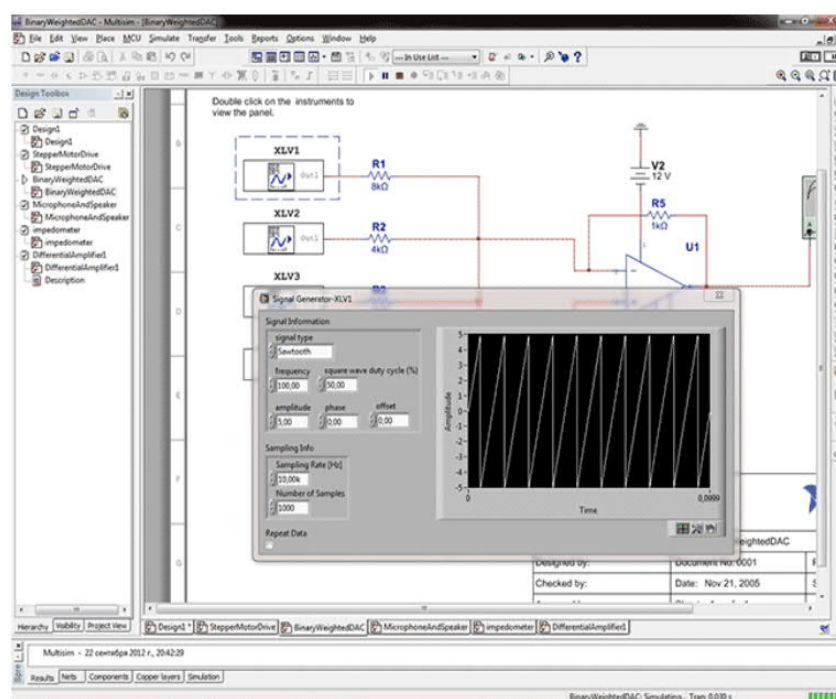


Рисунок 2.10 – Налаштування параметрів компонентів в NI Multisim

Для полегшення процесу створення друкованих плат компанія дає можливість кожному розробнику вступити в онлайн–спільнота NI Circuit Design Community для того, щоб обмінюватися своїми роботами, прототипами, шаблонами, обговорювати нюанси розробок і отримувати нові знання від колег і однодумців, які живуть по всьому світу [27].

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

3 АПАРАТНА РЕАЛІЗАЦІЯ МЕТОДУ НАСКРІЗНОГО ЗАПИСУ

3.1 Обґрунтування та налаштування елементів пам'яті в режимі запису інформації

Для розробки пристрою наскрізного запису інформації знадобляться два основні компоненти: регістр та дешифратор.

Було обрано регістр 74LS377, що зображений на рисунку 3.1, а таблиця 3.1 є його таблицею істинності. Він є 8-бітним регістром, побудованим з використанням передової технології Schottky з малою потужністю. Цей регістр складається з восьми тригерів типу D з буферизованим загальним годинником і буферизованим загальним входом. 74LS377 складається з восьми тригерів з тримачами країв з індивідуальними входами D і виходами Q. Годинник (CLK) і вхід EN_G є загальними для всіх тригерів. Коли EN_G є низьким, нові дані вводяться в регістр на зміні тактів з меншого до більшого CLK. Коли EN_G є високим, регістр зберігатиме дані, незалежно від стану CLK.

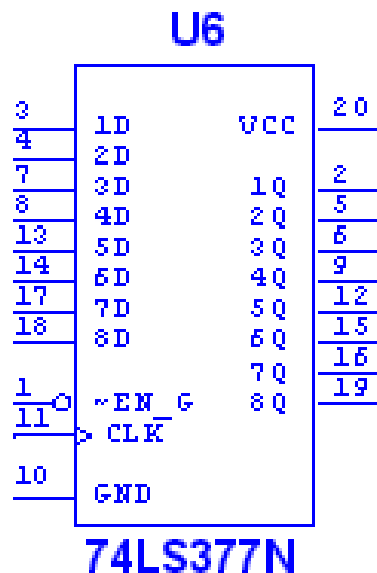


Рисунок 3.1 – Регістр 74LS377N – а) та його таблиця істинності – б)

Таблиця 3.1 – таблиця істинності регістра 74LS377N

Виходи			Вихід	Режим
EN_G	CLR	D1–D8	Q _i	
1	x	x	Q _i	Заборона запису
0	0	D	D	Запис
0	1	x	Q _i	Зберігання

Щоб перевірити правильність роботи регістра до нього було під'єднано чотири керуючих сигнали до інформаційних входів, і також ще два під'єднали до керуючих входів. На виході для перевірки правильності роботи було поставлено LED1. Результат зберігання можна побачити на рисунку 3.2.

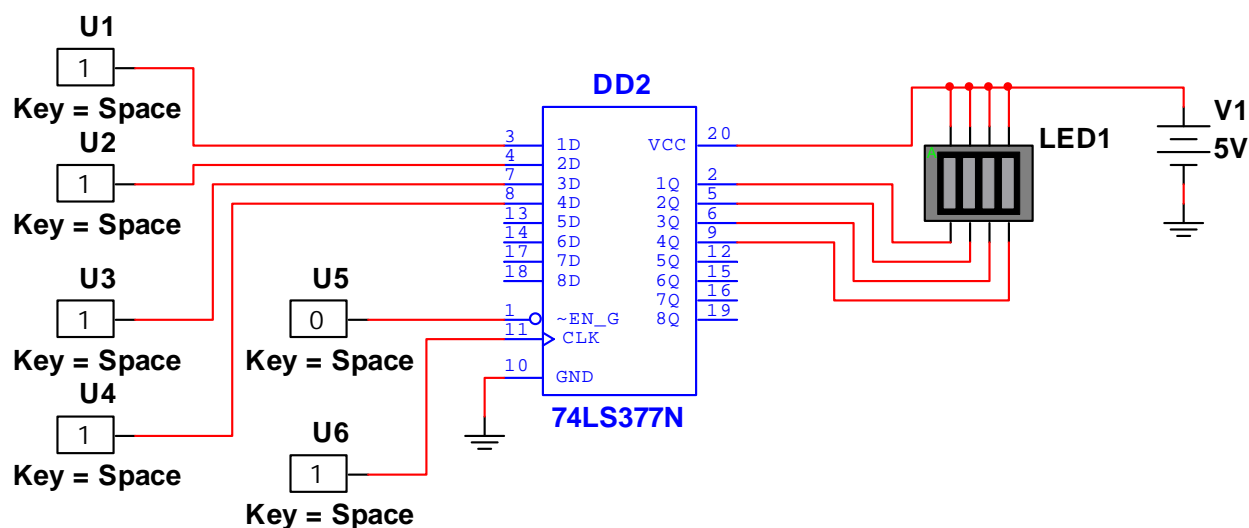


Рисунок 3.2 – Режим запису інформації в регістр 74LS377

В якості дешифратора нами було вибрано дешифратор 74154N (рисунок 3.3). В основі дешифратора використовують TTL-схеми для декодування чотирьох двійкових входів у один з шістнадцяти взаємно виключних виходів, коли обидва входи G1 і G2 є низькими.

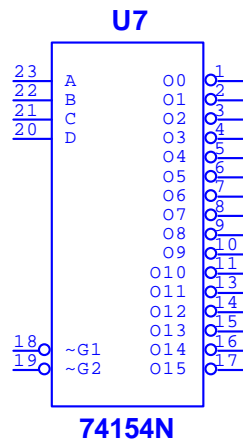


Рисунок 3.3 – Дешифратор 74154N

Принцип роботи дешифратора такий: якщо на вхоч G1 і G2 сигнали є низькими тоді відбувається процес дешифування з двійкового коду в шістнадцятковий код, якщо ж на одному із входів або ж на кожному з цих входів зміниться стан на високий, тоді усі вихідні сигнали змінять свій стан на високий.

Для перевірки праильновсті роботи дешифратора, до його входів було приєднано керуючі сигнали, а на виходи LED (це зображено на рисунку 3.5). Длє перевірки роботи ми дешифрували код “1100”, в результаті чого загорівся правильний третій вихід.

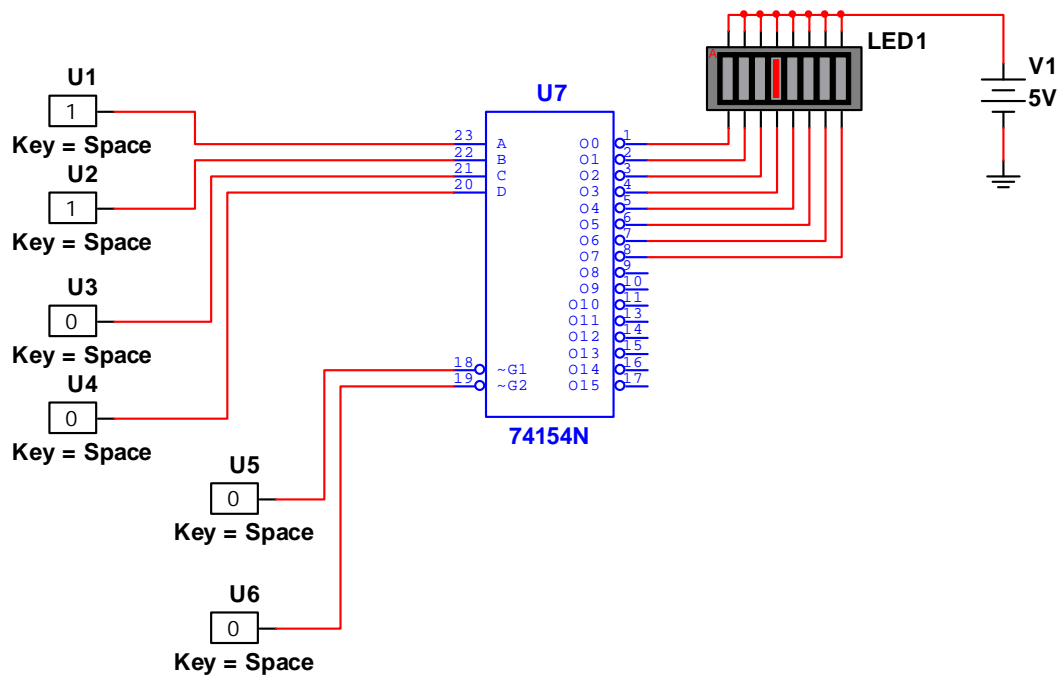


Рисунок 3.5 – Дешифрація вхідного коду 1100

На основі цих елементів з урахування логічних АБО, було створено скорочену систему запису інформації тільки з першим та останнім регістром пам'яті, що зображена на рисунку 3.6.

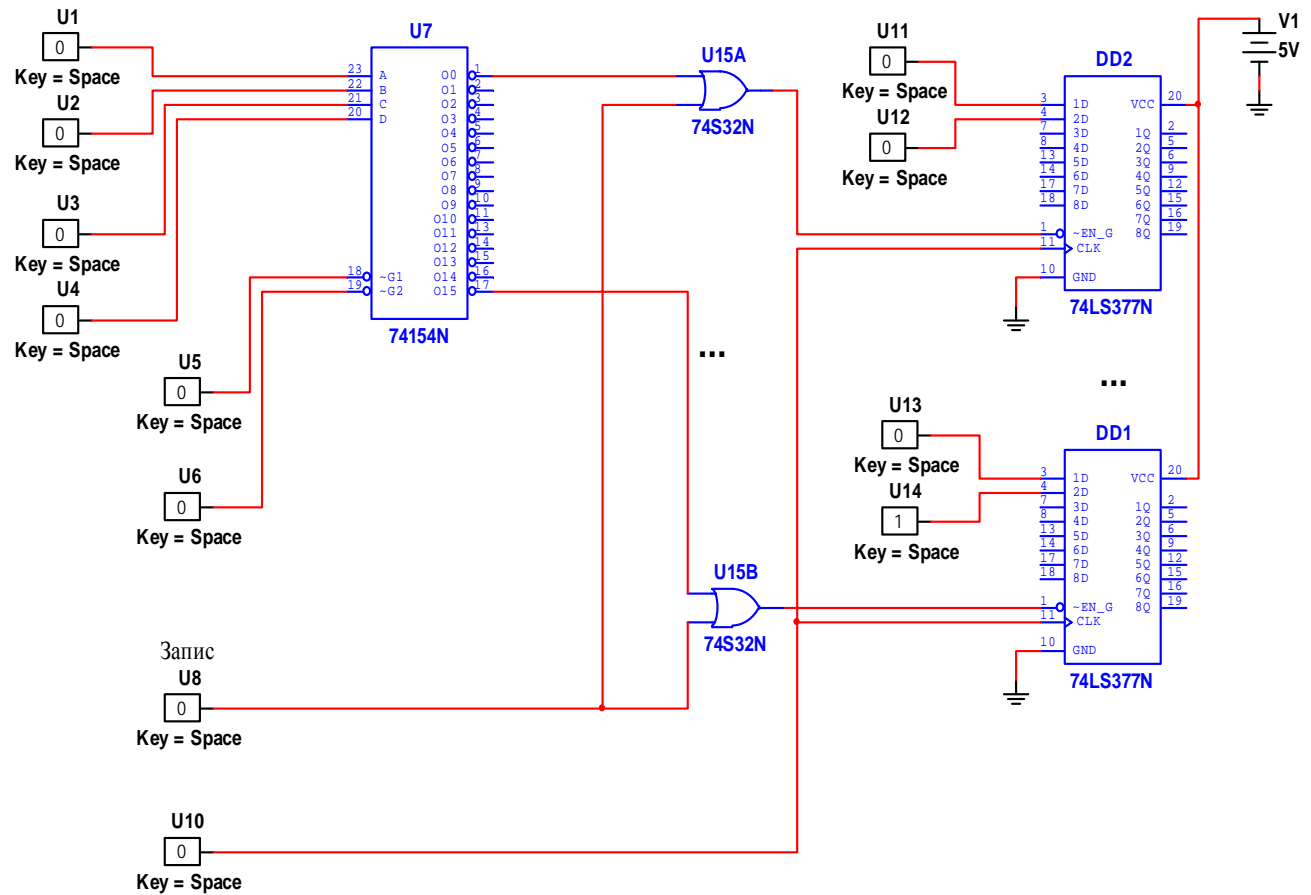


Рисунок 3.6 – Схема для запису інформації в регістри

По схемі видно, при подачі коду адреси інформація дешифрується і поступає на один з логічних елементів АБО де вона з'єднується через елемент з сигналом який дозволяє або забороняє запис інформації. Далі, якщо запис дозволений, то інформація записується в потрібному регістрі.

Для перевірки правильності роботи нашої схеми, до виходів регістрів було під'єднано LED1 і LED2, що зображено на рисунку 3.7. Записавши адресу, вона дешифрувалася і знаходила потрібний регістр, а при дозволяючому на сапис сигналу змінювалася інформація в регістрах. Якщо ж дозволяючого сигналу не було, або ж дешифрована адреса не відповідала регістру, то інформація в регістрі не змінювалася.

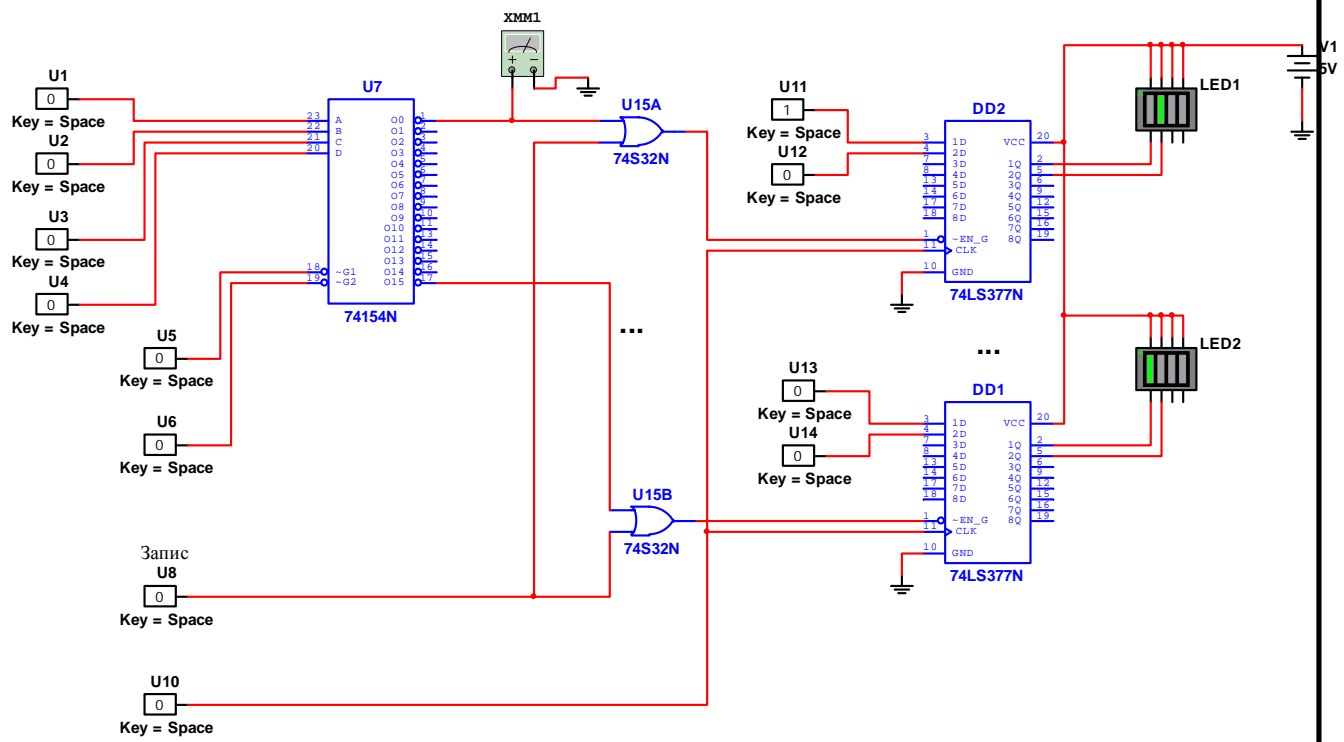


Рисунок 3.7 – Дослідження схеми запису інформації в регістри

Схема на рисунку 3.7 буде основою для розробки пристрою наскрізного запису інформації в Кеш та ОЗП.

Таким чином, для розробки схеми було вибрано основні елементи схеми: регістр 74LS377 та дешифратор 74154N. В цьому пункті було вивчено принцип роботи цих елементів, а також було проведено налаштування та перевірку їхньої роботи. Додавши логічні елементи АБО 75S32N, було розроблено та досліджено схему їхньої взаємодії під час запису інформації.

3.2 Розробка та налаштування контролера управління наскрізним записом

В обчислювальній техніці до одної шини необхідно підключати багато елементів. Однак в будь-який проміжок часу через дану шину будуть обмінюватися інформацією тільки два пристрої. Разом з тим пасивне підключення інших елементів пристроїв, що не беруть участь у комунікації буде

суттєво збільшувати сумарний струм шини. Це може спричинити такий спад напруги в шині, коли значення логічної 1 на ній буде неможливо. Крім цього необхідно посилити струм, що йде від джерела до приймача інформації. Ці завдання можна вирішити за допомогою елементів з трьома станами. На основі елементів з трьома станами виготовляються шинні формувачі. Для моєї схеми мені знадобиться шинний формувач 74ALS245AN, який зображений на рисунку 3.8 та його таблиця істинності (таблиця 3.2).

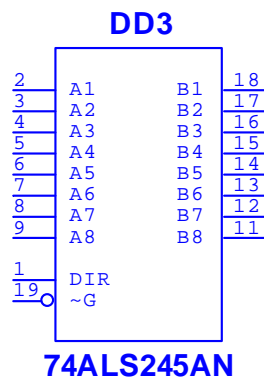


Рисунок 3.8 – Шинний формувач 74ALS245AN

Таблиця 3.2 – таблиця істинності шинного формувача 74ALS245AN

INPUTS		OUTPUTS
G	DIR	
L	L	Bus B data to Bus A
L	H	Bus A data to Bus B
H	X	Z

74ALS245A є вісімковим приймачем, що містить неінвертувальні трьохстанні сумісні виходи в напрямках передачі і прийому. Пристрій має вхід (G) для встановлення Z-стану і передачі/прийому (DIR) для керування напрямком. Z-стан характеризує собою стан при якому шинний формувач не приймає і не видає сигнали.

Для перевірки правильності роботи формувача я під'єднав до А1–А4 керуючі сигнали, а їм відповідним В1–В4 під'єднав LED1, щоб перевірити роботу в напрямку А до В. Натомість до А5–А8 під'єднав LED2, а їм відповідним В5–В8 їх керуючі сигнали. Результати перевірки виводу коду “1010” в напрямку з В до А показані на рисунку 3.9.

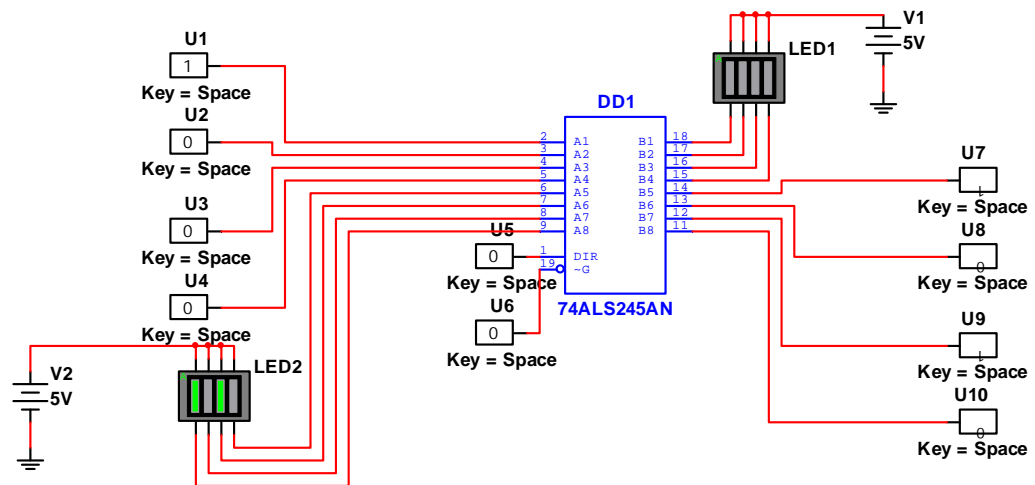


Рисунок 3.9 – Дослідження режиму передачі інформації з виводів В до А

Змінивши керуючий сигнала DIR, я перевіряв роботу в зворотньому напрямку, що показано на рисунку 3.10. Для перевірки я на входи А1–А4 подав сигнал “1001”, а на виходах В1–В4 отримав правильний результат.

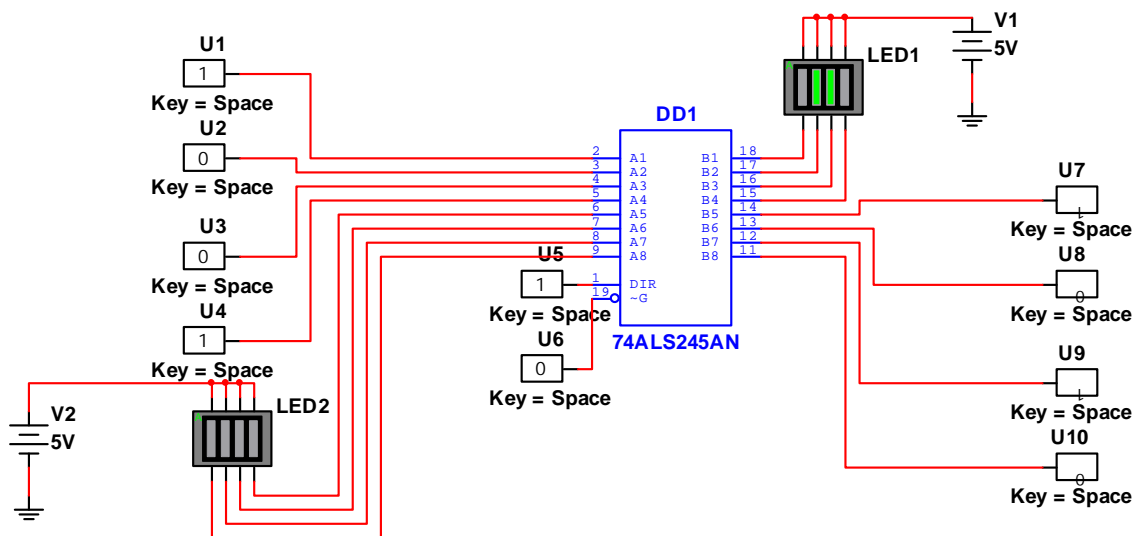


Рисунок 3.10 – Дослідження режиму передачі інформації з виводів А до В

Дослідивши павильність роботи шинних формувачів, було вирішено розробити схему, яка керується принципом наскрізного запису. Ця схема розміщена на рисунку 3.11. Розроблена схема була розроблена на основі схеми, що зображена на рисунку 3.6. Головна модифікація цієї схеми – це включення в схему тригера. На вхід D тригера поступає сигнал, який вказує, що є присутня або відсутня якась дешифрована адреса. До входу CLK поступають сигнали CLK# та ADS# (цей сигнал є інвертований), що об'єднані логікою. Для скидання тригера, на вхід тригера CLR прямують сигнали CLK#, ADS#, DRDY# та DBSY#, які теж об'єднані логікою. В розробленій схемі ми перевірили правильність роботи сигналу скидання та дозволяючого сигналу тригера за допомогою світлодіодів.

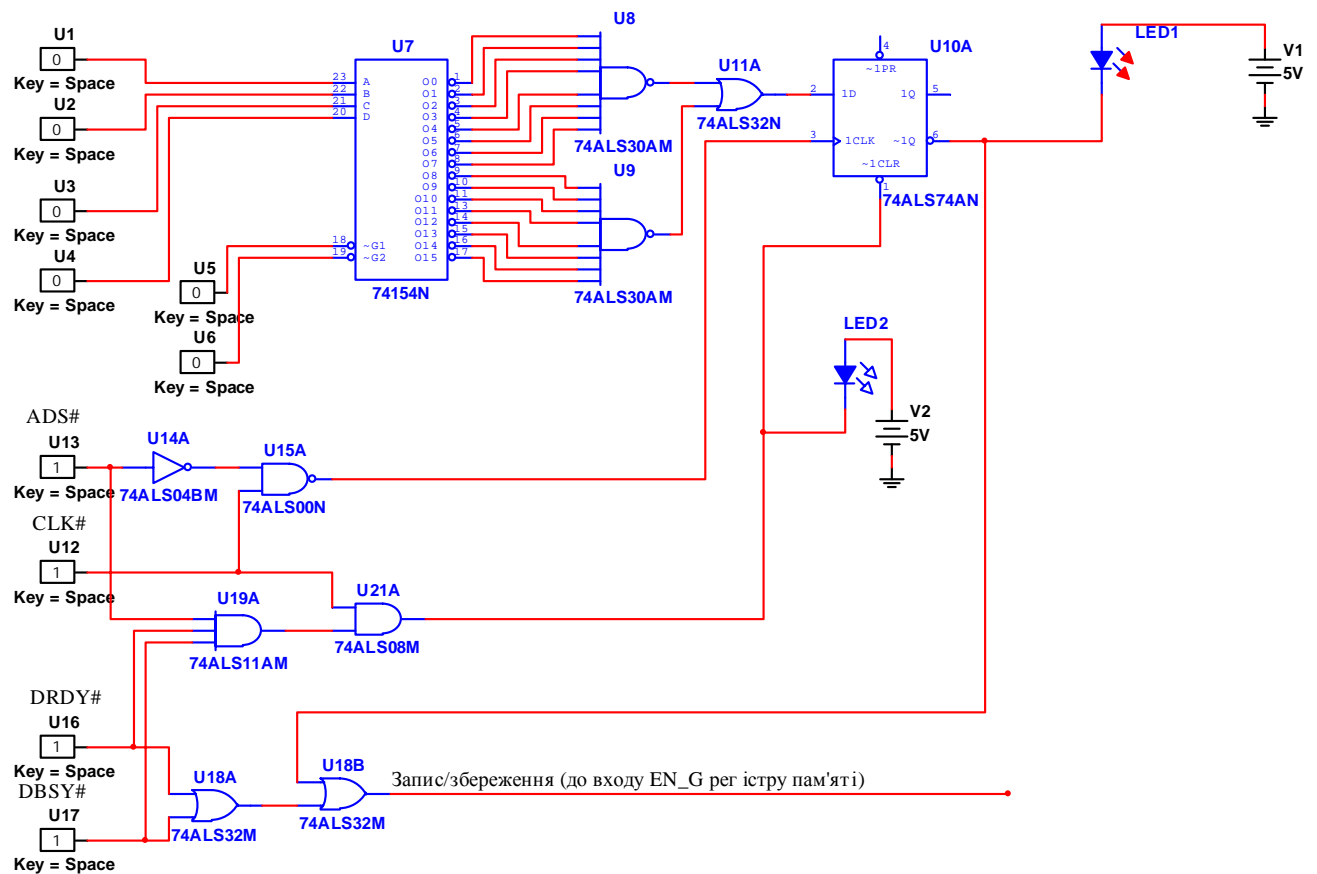


Рисунок 3.11 – Схема управління записом

Сигнал ADS# є дозволяючим приймання адрес, тому для режиму запису обов'язково, щоб він перебував в стані логічної 1. Завдання сигналів DRDY# та

					Арк.
					42
Зм.	Арк.	№ докум.	Підпис	Дата	БР.КСМ. 07110/15.00.00.000 ПЗ

DBSY# є те, що вони дозволяють передачу даних по системній шині регістрам. Для того щоб в тригері відбувся режим скидання один з цих сигналів (ADS#, DRDY# та DBSY#) або ж всі мають дорівнювати 0. Якщо мав відбутися режим скидання тоді загорівся LED2 і адреси не опрацьовував дані з дешифратора. Якщо ж усі ці сигнали дорівнюють 1, то на тригер поступали дані з дешифратора, і якщо був казана правильна адреса на виході \bar{Q} буде світитися LED1 (що й зображено на рисунку). В разі цього цей сигнал прямує до логічного АБО, де об'єднується уже з об'єднаними висналами DRDY# та DBSY#.

На основі цієї схеми було розроблено схему, що зображена на рисунку 3.12. До цієї схеми було добавлено регістри, а також шинні формувачі. Було вирішено не підключати в схему всі регістри та шинні формувачі, щоб не загроможувати і так відносно велику схему.

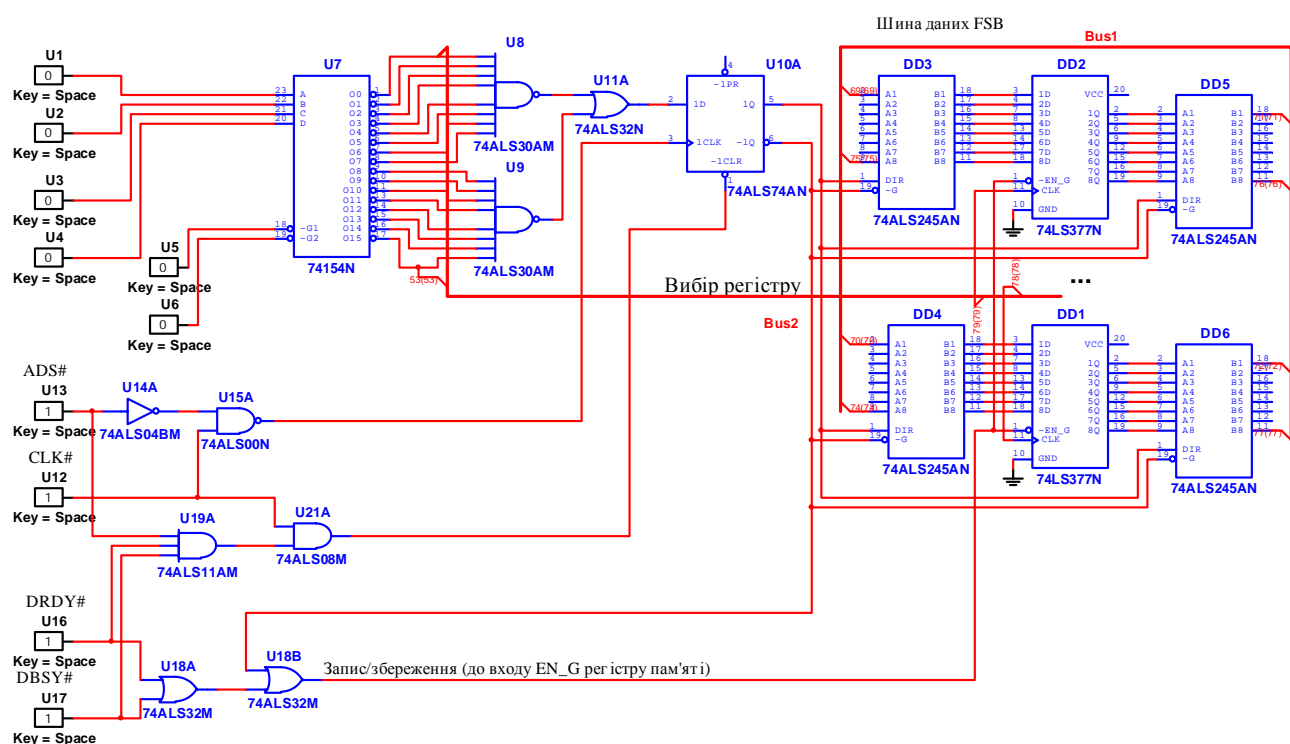


Рисунок 3.12 – Схема забезпечення наскрізного запису

Для забезпечення запису в схему будуть підключені дві шини. Шина даних FSB використовується для передачі даних в регістри, яка з'єднана з шинними формувачами. Шинні формувачі в свою чергу керуються сигналами з тригера,

що дозволяють або забороняють їм передачу даних в регістр. Натомість шина вибору регістра з'єднана з дешифратором для визначення потрібного регістру, а також з'єднана безпосередньо зі всіма регістрами, щоб дозволити одному з них сприйняти дані. Для дозволу запису інформації також використовується сигнал «Запису/збереження (до входу EN_G регістру пам'яті)» (сигнал є зображений на рисунку і включає в собі об'єднання сигналів DRDY#, DBSY# та вихода тригера \bar{Q}).

Таким чином, на основі доступних даних та попередньо розроблених схем, було створено спочатку схему управління записом. Під'єднавши до цієї схеми регістри та шинні формувачі я отримав схему забезпечення системи наскрізного запису. Реалізація схеми пройшла успішно і схема готова до роботи.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

4 ТЕХНІКО–ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ПРИСТРОЮ

4.1 Стадії технологічного процесу

У розробці проектного рішення задіяні наступні спеціалісти – розробники, а саме: керівник проекту (*К*); студент (*С*); консультант техніко–економічного розділу (КТЕО).

Форму поділу робіт по всіх основних етапах і видах робіт, які повинні бути виконані показано в таблиці 4.1.

Таблиця 4.1 – Середній час виконання проекту та стадії технологічного процесу

Назва операції (стадії)	Виконавець, посада	Середній час виконання операції, год.
Підготовка	Студент	7
Розробка проекту системи забезпечення наскрізного запису	Керівник ДП	16
	Консультант ТЕО, доцент	2
	Студент	219
Проектування технічної частини	Студент	24
Встановлення та налаштування прогр. Зас.	Студент	8
Тестування	Студент	3
Разом		279

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування.

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{OP} = \sum_i^N \sum_j^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.1)$$

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

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

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

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0 (1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij}^0 – 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.).

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.2 – Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад (стипендія), грн.	Коефіцієнт Додаткової з/п
Керівник ДП, викладач	7293	0,47
Консультант техніко–економічного розділу, доцент	7293	0,47
Студент	1400	0

Середня годинна ставка керівника та консультанта техніко–економічного розділу ДП дорівнює:

$$C_{ij} = \frac{7293 \cdot (1 + 0,47)}{168} = 20,4 \text{ (грн/год)} \quad (4.3)$$

Середня годинна оплата студента дорівнює:

$$C_{ij} = \frac{1400}{168} = 8,33 \text{ (грн/год)} \quad (4.4)$$

Витрати на оплату праці складають:

$$B_{ОП} = 16 \cdot 20,4 + 2 \cdot 20,4 + 261 \cdot 8,33 = 2166,53 \text{ (грн)} \quad (4.5)$$

Крім того, слід визначити 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:

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

$$B_{\phi} = \frac{20,5}{100} \cdot 2166,53 = 444,14 \text{ (грн)} \quad (4.7)$$

4.3 Розрахунок матеріальних витрат

Матеріальні витрати це вартість витрачених матеріалів, малоцінних та швидкозношуваних предметів на виробництво продукції, робіт або послуг, а також матеріалів і МШП, витрачених на адміністративні, збутові та інші потреби підприємства.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.8)$$

де K_i – витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i – ціна за одиницю i -го типу матеріалу, грн.; i – тип матеріального ресурсу;

n – кількість типів матеріальних ресурсів.

Звідси, витрати на матеріальні ресурси дорівнюватимуть: 1744,9грн. Проведені розрахунки занесемо у таблицю 4.3.

Таблиця 4.3 – Зведені розрахунки покупних виробів

Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн.	Транспортні витрати (10% від суми)	Загальна сума, грн.
Дешифратор 74154N	шт.	1	5,7	5,7	0,57	6,27
Тригер 74ALS74AN	шт.	1	17	17	1,17	18,17

					БР.КСМ. 07110/15.00.00.000 ПЗ			Арк.
								48
Зм.	Арк.	№ докум.	Підпис	Дата				

Продовження таблиці 4.3

Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн.	Транспортні витрати (10% від суми)	Загальна сума, грн.
Регістр 74LS377N	шт.	16	11	176	17,6	193,6
Шинний формувач 74ALS245AN	шт.	32	18,8	601,6	60,16	661,76
74ALS30AM 8-INPUT NAND	шт.	2	6,25	12,5	1,25	13,75
74ALS32N 2-INPUT OR	шт.	3	14	42	4,2	46,2
74ALS08M 2-INPUT AND	шт.	1	5	5	0,5	5,5
74ALS00N 2-INPUT NAND	шт.	1	10	10	1	11
74ALS11AM 3-INPUT AND	шт.	1	7,9	7,9	0,79	8,69
74ALS04BM INVERTER	шт.	1	4,6	4,6	0,46	4,46
Паяльний набір	шт.	1	500	500	50	550
Папір (формат А4)	уп.	1	70	70	7	77
Тонер для принтера	уп.	1	135	135	13,5	148,5
Р а з о м						1744,9

4.4 Розрахунок витрат на електроенергію

Якщо для розробки КС використовується електрообладнання, то необхідно розрахувати витрати на електроенергію.

Загальна сума витрат на електроенергію розраховується за формулою:

$$B_E = \sum_{i=1}^n P_i \cdot k_i \cdot T_i \cdot Ц, \quad (4.9)$$

де P_i – паспортна потужність i -го електрообладнання, кВт;

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

k_i – коефіцієнт використання потужності i -го електрообладнання (приймається 0.7 , 0.9);

T_i – час роботи i -го устаткування за весь період розробки, год;

C – ціна електроенергії, грн / кВт· год;

i – тип електрообладнання;

n – кількість електрообладнання.

Для розробки дипломного проекту використовується один ноутбук потужністю $P = 0,5$ кВт, який за весь період розробки працює 100 годин, та друкуючий пристрій потужністю $P = 0,37$ кВт, який працює 2 години, і паяльник який працює 6 годин.

Проміжні розрахунки на витрату електроенергії подані в таблиці 4.4

Таблиця 4.4 – Витрати на електроенергію

Найменування устаткування	Паспортна потужність, кВт	Коефіцієнт використання потужності	Час роботи обладнання для розробки, год	Ціна електроенергії, грн/кВт·год	Сума, грн.
Ноутбук	0,5	0,9	100	0,9	40,5
Паяльник	0,7	0,9	6	0,9	3,78
Разом					44,28

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників. Накладні витрати для даного проекту подані далі.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

$$B_{\Phi} = 0,7 \cdot B_{ОП}, \quad (4.10)$$

де H_B – накладні витрати.

$$H_B = 0,7 \cdot 2166,53 = 1516,57 \quad (4.11)$$

4.6 Інші витрати

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

$$I = 3815,88 \cdot 0,1 = 216,65 \quad (4.12)$$

4.7 Складання кошторису витрат та визначення собівартості

Загальні витрати ($K_{ІПР}$) розрахуємо за формулою:

$$K_{ІПР} = B_{ОП} + B_{\Phi} + B_M + B_{ЕЛ} + H_B + I \quad (4.13)$$

Тобто:

$$K_{ІПР} = 6133,07 \quad (4.14)$$

Результати проведених розрахунків зведемо у таблицю 4.5.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 4.5 – Кошторис витрат

Зміст витрат	Сума, грн.
Витрати на оплату праці (осн. і дод. ЗП)	2166,53
Відрахування на соціальні заходи	444,14
Матеріальні витрати	1744,9
Витрати на електроенергію	44,28
Накладні витрати	1516,57
Інші витрати	216,65
Разом	6133,07

4.8 Розрахунок ціни проекту

Договірна ціна (C_D) для проектних рішень розраховується за формулою:

$$C_D = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.15)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p – середній рівень рентабельності, % (приймаємо 30% за погодженням з керівником).

$$C_D = 9440,03 \cdot 1,3 = 7972,99 \text{ (грн.)} \quad (4.16)$$

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

4.9 Визначення економічної ефективності і терміну окупності капітальних вкладень

Економічна ефективність – досягнення найбільших результатів за найменших затрат живої та уречевленої праці. Економічна ефективність є конкретною формою дії закону економії часу. За капіталістичного способу виробництва узагальнюючий показник економічної ефективності — норма прибутку.

Економічна ефективність – це співвідношення між отриманим прибутком та затраченими коштами. Вона обчислюється за формулою (4.17):

$$E_{\phi} = \Pi_p / K_B \quad (4.17)$$

де Π_p – очікуваний прибуток ;

K_{IP} – кошторисна вартість.

Очікуваний прибуток можна розрахувати із співвідношення:

$$\Pi_p = Ц_d - K_{IP} = 7972,99 - 6133,07 = 1839,93 \text{ (грн.)} \quad (4.18)$$

Після проведених розрахунків отримуємо:

$$E_{\phi} = 1839,93 / 7972,99 = 0,23 \quad (4.19)$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_{\phi}} \quad (4.20)$$

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

Тобто:

$$T_P = 1 / E_{\phi} = 1 / 0,23 = 4,3 \text{ (роки.)} \quad (4.21)$$

Прийнятним вважається термін окупності близький до 7 років. Розраховані економічні показники проекту занесемо до таблиці 4.6.

Таблиця 4.6 – Економічні показники розробки

Показник	Значення
Собівартість, грн.	6133,07
Плановий прибуток, грн.	1839,93
Ціна, грн.	7972,99
Економічна ефективність	0,23
Термін окупності, рік	4,3

Враховуючи основні економічні показники з таблиці 4.6, можна зробити висновок, що при економічній ефективності 0,23 та терміні окупності – 4,3 роки проводити роботи по впровадженню даної системи забезпечення наскрізного запису є доцільним та економічно вигідним. Тому, з метою зниження вартості, варто було б здійснювати закупівлю обладнання у офіційних дилерів вказаних марок обладнання.

ВИСНОВКИ

В ході роботи над бакалавратською роботою, мною було досліджено одну з найважливіших складових комп'ютера – його пам'ять. Було опрацьовано велику кількість інформації про пам'ять, визначено важливість саме ієрархічного принципу побудови пам'яті та кожної ланки цього принципу, зокрема ОЗП та кеш. Взаємодія цих двох ланок суттєво впливає на швидкість та правильність роботи самого комп'ютера. Якщо процесор взаємодіє безпосередньо з кешем, тоді швидкість не зменшується, тому потрібно щоб в кеші завжди була актуальна та потрібна інформація для процесора (нажаль це не завжди так). При записі даних, актуальність інформації може втрачатися, оскільки в кеші та ОЗП вона може відрізатися. Тому для забезпечення ідентичності цих даних використовують метод наскрізного запису.

На основі вивчених даних було розроблено структурну схему системи наскрізного запису в Кеш-пам'ять та основну пам'ять. Уже на основі цієї структурної схеми було вибрано основні елементи нашої схеми. До цих елементів ввійшли: регістр 74LS377, дешифратор 74154N та шинний формувач 74ALS245AN. Було вивчено їхні даташити, способи та режими їхньої роботи, а їхню роботу було перевірено в програмному забезпеченні NI Multisim 14.0, яке для мене уже було знайомим.

В процесі реалізації було розроблено чотири основні схеми, які теж були реалізовані в NI Multisim 14.0. Перша схема забезпечувала звичайний запис даних в регістри, а друга перевіряла правильність роботи. На основі першої та другої схеми було створено схему для управління записом, в якій важливу роль відігравав тригер. Перевіривши роботу схеми управління записом, на її основі було спроектовано схему забезпечення наскрізного запису додавши до неї шини, регістри та шинні формувачі.

В ході роботи над техніко-економічним розділом було обраховано можливі витрати і обраховано приблизну вартість пристрою забезпечення

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						55
Зм.	Арк.	№ докум.	Підпис	Дата		

наскрізного запису. Було виявлено, що схема є економічно ефективною. В додатку Б розміщена довідка про використання, що так само підтверджує якість пристрою. А в додатку А є тези з конференції, які були написані в результаті роботи над бакалавратською роботою.

Забезпечення ідентичності інформації КП та ОЗП є досить великою проблемою в комп'ютерній техніці оскільки це впливає на швидкість роботи комп'ютера, саме тому і актуальність цієї теми є великою. Забезпечення цієї ідентичності шляхом наскрізного запису є ефективним , через те, що спосіб наскрізного запису вирішує це питання зразу, оновивши дані зразу в кеші та основній пам'яті.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мельник А.О. Архітектура комп'ютера: навчальний посібник. Луцьк, 2008. С. 307–397.
2. Гук М. Ю. Аппаратные средства IBM PC. Энциклопедия. Санкт–Петербург, 2006. С. 334–398.
3. Жуйков В.Я., Терещенко Т.О., Ямненко Ю.С., Заграничний А.В. Мікропроцесорна техніка: навчальний посібник. Київ, 2016. 168 с.
4. Кеш-пам'ять та її роль в комп'ютерних системах: веб–сайт URL: <https://studfiles.net/preview/3741158/page:26/> (дата звернення: 12.02.19).
5. Забезпечення ідентичності вмісту блоків кеш пам'яті і основної пам'яті: веб–сайт. URL: <https://studfiles.net/preview/3907471/page:144/> (дата звернення: 13.02.19).
6. Електронна пам'ять в комп'ютері та її специфікації: веб–сайт. URL: https://studopedia.su/2_14286_elektronna-pamyat.html (дата звернення 13.02.19).
7. Оптимізація роботи оперативної пам'яті: веб сайт. URL: <https://www.br.com.ua/kurs/Computers/52199-1.html> (дата звернення 17.03.19).
8. Класифікація електронної пам'яті та її специфікації: веб–сайт. URL: https://studopedia.com.ua/1_24947_klasifikatsiya-elektronnoi-pamyati.html (дата звернення: 17.02.19).
9. Типи пам'яті. Порядок контролю парності і логічної організації пам'яті. Коди корекції помилок: веб–сайт. URL: <http://www.wikipage.com.ua/1x8f9f.html> (дата звернення: 18.03.19).
10. Архітектура персонального комп'ютера: веб–сайт. URL https://elearning.sumdu.edu.ua/free_content/lectured:1a259358378153792bb8645df287e86d790fc40d/20160903092057/44894/index.html (дата звернення: 18.03.19).
11. Структура оперативної пам'яті: веб–сайт. URL: https://studopedia.su/2_14287_shvidkodiya-i-produktivnist-pamyati.html: (дата звернення: 2.05.19).

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

12. Постійний запам'ятовуючий пристрій: веб-сайт. URL: <https://studfiles.net/preview/3907471/page:128/> (дата звернення: 5.05.19)

13. Кеш-пам'ять в складі сучасного комп'ютера її характеристики: веб-сайт. URL: <http://ua-referat.com/%9A%D0%B5%D1%8D0%9A%D0%B5%D1%88-%D0%BF%D0%B0%D0%BC%60%D1%8F%D1%82%D1%8C> (дата звернення: 10.02.19).

14. Оперативна (основна Main Memory) пам'ять (ОП): веб-сайт. URL: <http://um.co.ua/4/4-11/4-119672.html> (дата звернення: 5.05.19).

15. Взаємодія процесора з пам'яттю в комп'ютері з простою системою команд: веб-сайт. URL: <https://studfiles.net/preview/3907471/page:60/> (дата звернення: 10.05.19).

16. Значення кеш пам'яті. Як працює кеш пам'ять: веб-сайт. URL: <https://gadgetsroom.ru/the-value-of-the-cache-memory-how-the-cache-works/> (дата звернення: 10.05.19).

17. Верхній рівень структурної організації комп'ютера. Як працює кеш пам'ять: веб-сайт. URL: <http://lib.exdat.com/docs/756/index-859-16.html?page=3> (дата звернення: 10.05.19).

18. Керування пам'яттю комп'ютера та взаємодія всіх її ланок: веб-сайт. URL: <http://pzks.nmu.org.ua/ua/labs/os5.pdf> (дата звернення: 10.04.19).

19. Кешування пам'яті: веб-сайт. URL: <https://biglib.info/5434-keshuvannya-pamyat.html> (дата звернення: 18.05.19).

20. Принципи та способи організації Кеш-пам'яті в комп'ютері: веб-сайт. URL: <https://studopedia.org/6-105219.html> (дата звернення: 10.05.19).

21. Організація пам'яті процесора та його взаємодія з основною пам'яттю: веб-сайт. URL: <https://homework.net.ua/organizatsiya-pam-yati-protsesora/> (дата звернення: 10.05.19).

22. Організація обміну інформацією між процесором і основною пам'яттю через кеш пам'ять: веб-сайт. URL: <https://studfiles.net/preview/71/page:3/> (дата звернення: 10.04.19).

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

23. An FPGA Approach to Quantifying Coherence Traffic Efficiency on Multiprocessor Systems: веб-сайт. URL: https://www.researchgate.net/publication/4290181_An_FPGA_Approach_to_Quantifying_Coherence_Traffic_Efficiency_on_Multiprocessor_Systems (дата звернення: 10.04.19).

24. Пирамидальный дешифратор и его особенности: веб-сайт. URL: <https://www.ngpedia.ru/id6p3.html> (дата звернення: 10.05.19).

25. Дешифратори. Класифікація: веб-сайт. URL: <https://studopedia.org/2-150638.html> (дата звернення: 10.04.19).

26. NI Multisim: веб-сайт. URL: <http://cxem.net/software/multisim.php> (дата звернення: 10.04.19).

27. NI Multisim: веб-сайт. URL: https://en.wikipedia.org/wiki/NI_Multisim (дата звернення: 10.04.19).

28. 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.

29. 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.

30. Методичні вказівки до написання техніко-економічного розділу дипломних проектів освітньо-кваліфікаційного рівня «бакалавр» підготовки 6.050102 комп'ютерна інженерія/ І.Р. Паздрій Тернопіль: ТАНГ, 2014. 37 с.

31. Гросуляк П.І. Методи організації обміну інформації між case- та оперативною пам'яттю / Гросуляк П.І., Шкодич О.В. // Тези доповідей Науково-практичної конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». ч.2 – Тернопіль: ТНЕУ. – 2019. – С. 16.

					БР.КСМ. 07110/15.00.00.000 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		