

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

БАРАН Євген Романович

**Надоперативна пам'ять на основі архітектури
кластерного розподіленого реєстрового
файлу / Excessive memory based on the
architecture of a clustered distributed registry
file**

Спеціальність: 123 – Комп'ютерна інженерія
Освітньо-професійна програма – Комп'ютерна інженерія

Випускна кваліфікаційна робота

Виконав: студент групи КСМ-43/2
БАРАН Євген Романович

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

ТЕРНОПІЛЬ -2019

РЕЗЮМЕ

Дипломний проект містить 61 сторінок пояснюючої записки, 17 рисунків, 8 таблиць, 1 додаток. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломного проекту є апаратна реалізація пристрою надоперативної пам'яті на основі архітектури кластерного розподіленого реєстрового файлу.

Методи досліджень – методи побудови моделей та електричних схем цифрових пристроїв.

Проведено аналіз методів побудови апаратно-програмних систем захисту показав трудомісткість, але разом з тим перспективність апаратної реалізації. Здійснено обґрунтування вибору напрямку дослідження та здійснена постановка задачі постановка задачі Деталізовано процеси захисту інформації в криптосистемі Рабіна;

Розроблено структуру апаратного модуля дешифрації та її основних компонентів. Здійснено апаратну реалізацію схеми та верифікацію електричних схем знаходження модуля числа, підкореневого виразу та знаходження кореня;

.Ключові слова: МІКРОСХЕМА, СТРУКТУРНА СХЕМА, ЕЛЕКТРИЧНА СХЕМА, КОМБІНАЦІЙНА ЛОГІКА, ШИФРАТОР, ЛІЧИЛЬНИК, МУЛЬТИПЛЕКСОР, ПРОГРАМОВАНА ЛОГІЧНА МАТРИЦЯ, КОНТРОЛЕР.

SUMMARY

The diploma project contains 61 pages of explanatory note, 17 figures, 8 tables, 1 appendix. Volume of graphic material 2 sheets of A3 format.

The purpose of the diploma project is the hardware implementation of the superoperative memory device based on the architecture of the cluster distributed register file.

Research methods - methods of building models and electrical circuits of digital devices.

The analysis of methods of construction of hardware and software protection systems showed the complexity, but at the same time the prospects of hardware implementation. The substantiation of the choice of the direction of research is carried out and the statement of the problem is made the statement of the problem The processes of information protection in the Rabin cryptosystem are detailed;

The structure of the hardware decryption module and its main components is developed. Hardware implementation of the circuit and verification of electrical circuits of finding the modulus of a number, subroot expression and finding the root.

Key words: MICROSCHHEME, STRUCTURAL DIAGRAM, ELECTRICAL DIAGRAM, COMBINATION LOGIC, CODE, COUNTER, MULTIPLEXOR, SOFTWARE.

ЗМІСТ

Вступ	9
1 Принципи побудови та апаратна реалізація засобів оперативної пам'яті	11
1.1 Особливості застосування оперативної пам'яті	11
1.2 Принципи функціонування пристроїв надоперативної пам'яті	15
1.3 Постановка задачі по реалізації пристрою	18
2 Проектування компонентів пристрою	20
2.1 Методики побудови пристроїв реєстрової пам'яті	20
2.2 Проектування структури модуля реєстрової пам'яті на основі кластерного підходу	23
2.3 Структура контролера управління кластерним реєстровим файлом	26
2.4 Обґрунтування вибору системи автоматизованого проектування пристрою	30
3 Апаратна реалізація та верифікація пристрою надоперативної пам'яті .	36
3.1 Розробка схеми контролера інтерфейсу та його налаштування	39
3.2 Апаратна реалізація схеми кластерного розподіленого реєстрового файлу	42
4 Техніко-економічний розділ	42
4.1 Розрахунок витрат на розробку апаратного модуля	48
4.2 Визначення експлуатаційних витрат	49
4.3 Розрахунок зведених економічних показників	51
Висновки	52
Список використаних джерел	55
Додаток А Довідка про використання	

					БР.КСМ.07159/17.00.00.000ПЗ			
<i>Зм.</i>	<i>Арк</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	НАДОПЕРАТИВНА ПАМ'ЯТЬ НА ОСНОВІ АРХІТЕКТУРИ КЛАСТЕРНОГО РОЗПОДІЛЕНОГО РЕГІСТРОВОГО ФАЙЛУ	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
Розробив		Баран С.Р.						
Перевірив		Масляк Б.О.						
Консульт.		Паздрій І.Р.						
Н. Контр.		Гураль І.В.						
Затв.		Березький О.				ТНЕУ, ФКІТ, КСМ-43/2		

ВСТУП

Одним з важливих чинників що підвищують продуктивність процесора, є наявність надоперативної статичної пам'яті, а точніше її обсяг, швидкість доступу і розподіл за рівнями. Вже досить давно практично всі процесори оснащуються даним типом пам'яті.

Статична пам'ять (Static Random Access Memory, SRAM), як і випливає з її назви, здатна зберігати інформацію в статичному режимі - тобто як завгодно довго при відсутності звернень (але при наявності напруги живлення). Елементи статичної пам'яті реалізуються на тригерах - елементах з двома стійкими станами. У порівнянні з динамічною пам'яттю ці комірки більш складні і займають більше місця на кристалі, проте вони забезпечують значно вищу швидкодію і використовуються в модулях кеш-пам'яті. Саме тому мікросхеми кеш-пам'яті є значно дорожчими, при цьому маючи невеликі обсяги комірок. Але з такої структури, випливає головна перевага такої пам'яті - швидкість. Так як тригери не потребують регенерації, а час затримки вентиля, на яких вони зібрані, невеликий, то час перемикання тригера з одного стану в інший відбувається дуже швидко. Це дозволяє кеш-пам'яті працювати на таких же частотах, що і сучасні процесори.

Кеш-пам'ять - це надшвидка пам'ять, яка використовується процесором, для тимчасового зберігання даних, які найбільш часто використовуються.

Кеш-пам'ять (cache memory) - є буфером між ОЗУ і її «клієнтами» - процесором (одним або декількома) і іншими абонентами системної шини. Кеш-пам'ять не є самостійним сховищем; інформація в ній не адресується клієнтами підсистеми пам'яті, присутність кеша для них «прозора». Кеш зберігає копії блоків даних тих областей оперативного запам'ятовуючого пристрою (ОЗП), до яких відбувалися останні звернення, і дуже ймовірно подальше звернення до тих же даних буде обслуговано кеш-пам'яттю істотно швидше, ніж оперативною пам'яттю. Від ефективності алгоритму кешування

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

залежить вірогідність знаходження затребуваних даних в кеш-пам'яті і, отже, виграш в продуктивності пам'яті і комп'ютера в цілому.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

1 ПРИНЦИПИ ПОБУДОВИ ТА АПАРАТНА РЕАЛІЗАЦІЯ ЗАСОБІВ ОПЕРАТИВНОЇ ПАМ'ЯТІ

1.1 Особливості застосування оперативної пам'яті

Існують різні підходи до аналізу архітектур комп'ютерної техніки. Популярним підходом є класифікація Флінна, яка передбачає організацію обчислювального процесу на основі паралелізму. З точки зору апаратної реалізації доцільніше розглядати архітектуру комп'ютера на основі аналізу зв'язку між його функціональними вузлами. У відповідності з даним підходом розрізняють одно- та двошинну архітектури. Введення інформації з пристрою введення до основної пам'яті, та виведення інформації з основної пам'яті до пристрою виведення здійснюється через процесор. В одношинній структурі комп'ютера (рисунок 1.1) всі його функціональні вузли з'єднані через одну спільну шину. Шина розподіляється між ними в часі, тому одночасно посилати дані на шину може тільки один вузол. Для узгодження швидкодії повільних і швидких вузлів комп'ютера в одношинній структурі на виході кожного повільного вузла (в даному випадку пристрої введення та виведення) використовують буферну пам'ять, яка дозволяє швидкий обмін блоками даних.

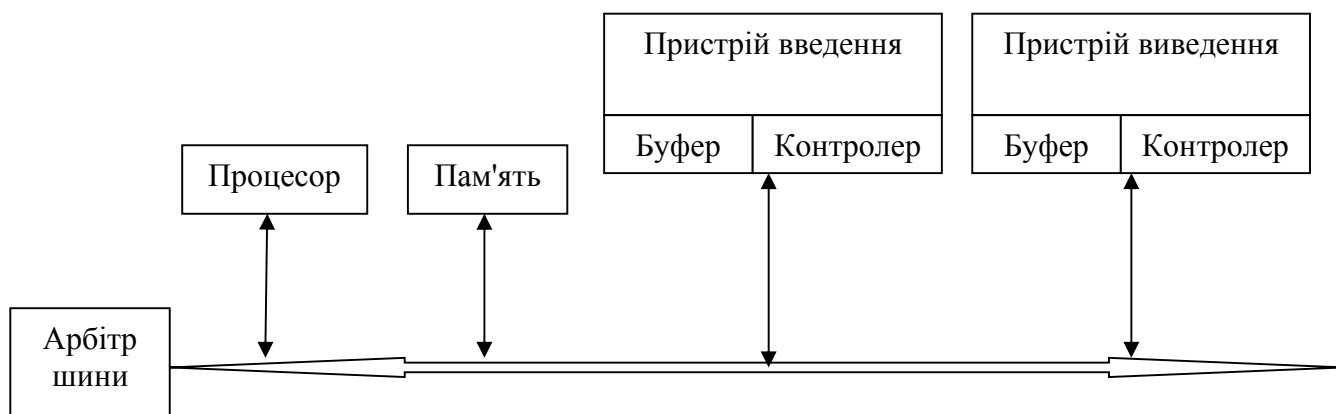


Рисунок 1.1 - Одношинна архітектура

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

Кожний пристрій введення та виведення складається з двох частин - контролера та самого пристрою. Контролер є спеціалізованим комп'ютером, який керує своїм пристроєм та його доступом до спільної шини.

Недоліком цієї структури є те, що процесор витрачає свій час на виконання операцій введення та виведення інформації, безпосередньо працюючи з пристроями введення-виведення, які в більшості випадків є значно повільнішими порівняно з ним, що знижує ефективність роботи комп'ютера.

В двошинній структурі комп'ютера з обміном через пам'ять (рисунок 1.2) процесор звільнений від організації операцій введення-виведення інформації.

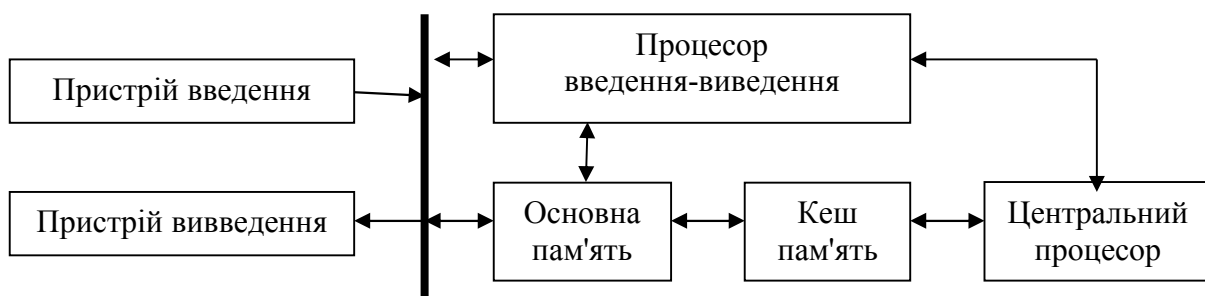


Рисунок 1.2 – Двошинна архітектура

Як видно з рисунків 1.1 та 1.2 незалежно від принципів організації обчислювальної системи важливим елементом комп'ютера є пам'ять. Пам'ять комп'ютера призначена для зберігання даних та програм. В комп'ютері використовуються різні типи пам'яті, які можуть бути класифіковані в залежності від способу доступу до інформації наступним чином:

- пам'ять з довільним доступом, що забезпечує запис або зчитування інформації за довільною адресою. За цим принципом побудовано регістрову пам'ять процесора та оперативну пам'ять;

- пам'ять із впорядкованим доступом, наприклад послідовним. За цим принципом побудовано буферну пам'ять, яка використовується у пристроях введення-виведення;
- пам'ять з асоціативним доступом в якій дані шукаються за їх змістом або за деякою їх ознакою (шаблон). За таким принципом побудовано кеш пам'ять.

Існує також пам'ять з прямим доступом. Кожен запис має унікальну адресу, що відображає її фізичне розміщення на носії інформації. Звернення здійснюється як адресний доступ до початку запису з подальшим послідовним доступом до певної одиниці інформації усередині запису. В результаті час доступу до певної позиції є величиною змінною. Такий режим характерний для магнітних дисків

Сучасними провідними фірмами випускається ряд пристроїв оперативної пам'яті у маркуванні яких зустрічаються букви U, E, R або F. Цими літерами позначається тип пам'яті - U (Unbuffered, небуферизована), E (пам'ять с корекцією помилок, ECC), R (регістрова пам'ять, Registered), F (FB-DIMM, Fully Buffered DIMM - повністю буферизована DIMM).

Небуферизовану пам'ять для звичайних настільних комп'ютерів називають UDIMM. На платі пам'яті, як правило, є 2, 4, 8 або 16 мікросхем пам'яті з однієї або двох сторін. У такий пам'яті маркування закінчується буквою U (Unbuffered), або взагалі без букви, наприклад DDR2 PC-6400, DDR2 PC-6400U, DDR3 PC-8500U або DDR3 PC-10600. А у пам'яті для ноутбуків маркування закінчується буквою S. Це скорочення від SO-DIMM, наприклад DDR2 PC-6400S – рисунок 1.1.

Пам'ять с корекцією помилок (ECC) є звичайною небуферизованою пам'яттю з корекцією помилок. Така пам'ять ставиться в фірмові (брендові) комп'ютери продаються в Європі (не сервери), плюсом цієї пам'яті є її велика надійність при роботі. Більшість помилок при роботі пам'яті вдається виправити під час роботи без втрати даних. На кожній платі такої пам'яті є 9 або 18 мікросхем пам'яті і додається одна або 2 мікросхеми.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

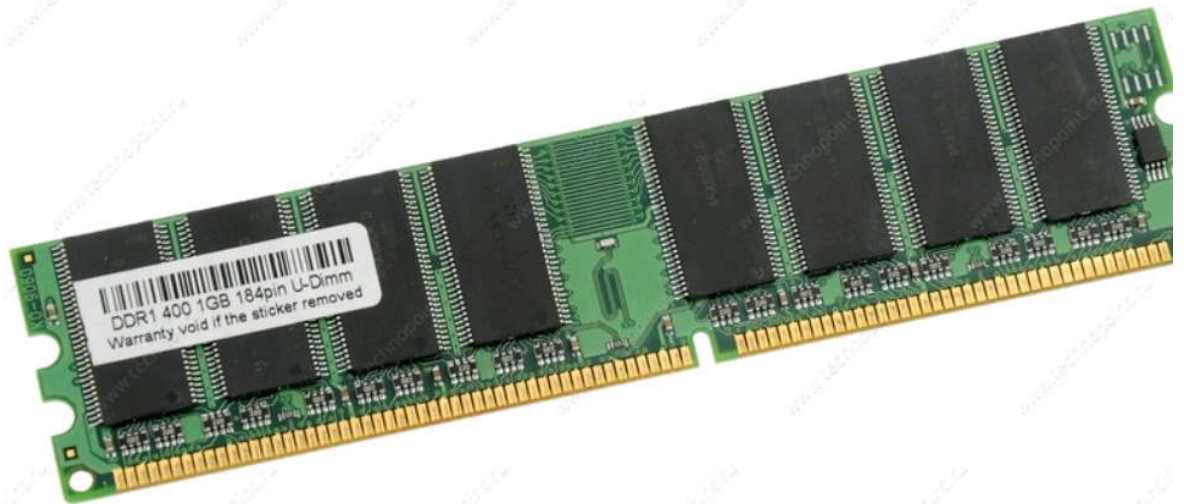


Рисунок 1.1 - Небуферизована пам'ять для персональних комп'ютерів
UDIMM

Більшість комп'ютерів і материнських плат можуть працювати з ECC пам'яттю. У такій пам'яті маркування як правило закінчується буквою E (ECC), наприклад DDR2 PC-4200E, DDR2 PC-6400E, DDR3 PC-8500E або DDR3 PC-10600E – рисунок 1.2.

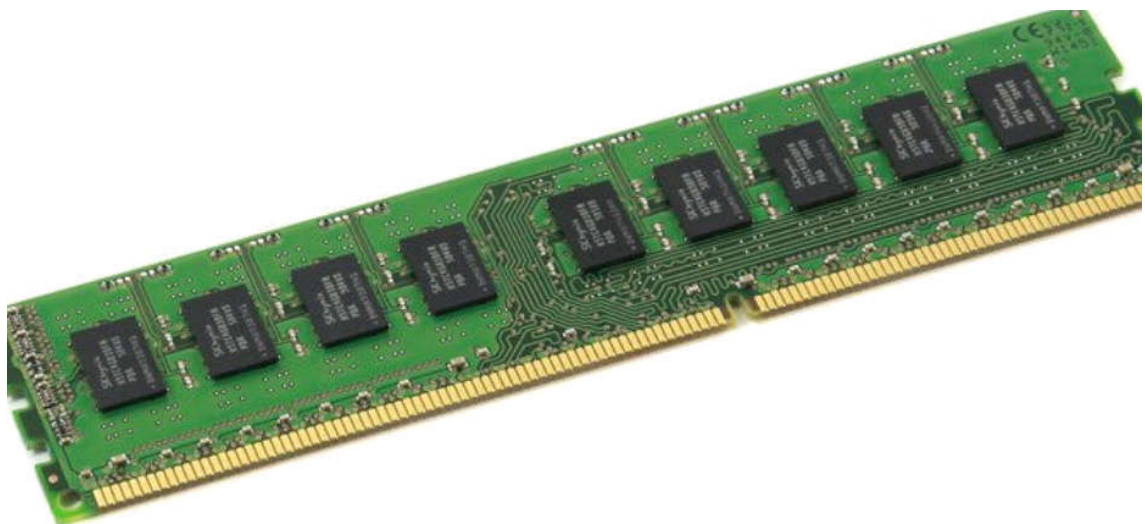


Рисунок 1.2 – Небуферизована пам'ять з корекцією помилок ECC

Різниця пам'яті з ECC і пам'яті без ECC можна бачити на рисунку 1.3.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

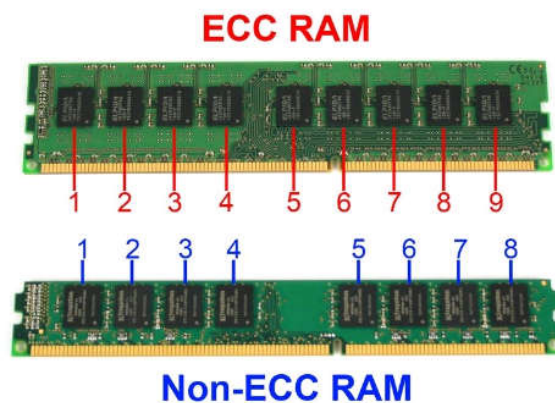


Рисунок 1.3 - Різниця пам'яті з ECC і пам'яті без ECC

Хоч більшість продаваних плат і підтримують цю пам'ять, але сумісність з конкретною платою і процесором краще дізнатися. Всі материнські плати і процесори починаючи з першого покоління Intel Core можуть працювати з ECC пам'яттю і від материнських плат це не залежить. Під AMD процесорами всі материнські плати можуть працювати з ECC пам'яттю, за винятком рідкісних випадків.

Регістрова пам'ять (англ. Registered Memory, RDIMM, іноді buffered memory) - вид комп'ютерної оперативної пам'яті, модулі якої містять регістр між мікросхемами пам'яті і системним контролером пам'яті. Наявність регістрів зменшує електричне навантаження на контролер і дозволяє встановлювати більше модулів пам'яті в одному пристрої. Регістрова пам'ять є більш дорогою через менший обсяг виробництва і наявності додаткових мікросхем. Регістрова пам'ять (Registered) є серверним типом пам'яті. Вона випускається з ECC (корекцією помилок) і є мікросхемою "Буфером". Мікросхема "буфер" дозволяє збільшити максимальну кількість плат пам'яті, які можна підключити до шини не перевантажуючи її. Останнім часом поняття буферизований і регістровий майже не розрізняють. Ця пам'ять працює тільки на серверних материнських платах здатних працювати з пам'яттю через мікросхему "буфер". На платах регістрової пам'яті з ECC встановлюється 9, 18 або 36 мікросхем пам'яті і ще 1, 2 або 4 мікросхеми

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		16

"буфера" (вони знаходяться в центрі плати і відрізняються за габаритами від мікросхем пам'яті). У такій пам'яті маркування закінчується буквою R (Registered), наприклад DDR2 PC-4200R, DDR2 PC-6400R, DDR3 PC-8500R або DDR3 PC-10600R. Ще в маркуванні реєстрової (серверної) пам'яті присутнє скорочення слова Registered – REG – рисунок 1.4.



Рисунок 1.4 – Серверна реєстрова пам'ять

Повністю буферизована пам'ять (FB-DIMM Fully Buffered DIMM) – є сучасним стандартом комп'ютерної пам'яті, який використовується для підвищення надійності, швидкості і обсягів підсистеми пам'яті. У традиційних стандартах пам'яті лінії даних підключаються від контролера пам'яті безпосередньо до ліній даних кожного модуля DRAM (іноді через буферні реєстри, по одній мікросхемі реєстра на 1-2 чіпа пам'яті). Зі збільшенням ширини каналу або швидкості передачі даних, якість сигналу на шині погіршується, ускладнюється разводка шини. Це обмежує швидкість і щільність пам'яті. FB-DIMM використовує інший підхід для вирішення цих проблем. Це подальший розвиток ідеї registered модулів - Advanced Memory Buffer здійснює буферизацію не тільки сигналів адреси, а й даних, і використовує послідовну шину до контролера пам'яті замість паралельної.

Модуль FB-DIMM має 240 контактів і однакову довжину з іншими модулями DDR DIMM, але відрізняється за формою виступів. Підходить тільки для серверних платформ.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

Таким чином, аналіз пристроїв комп'ютерної пам'яті показав їх значну відмінність в залежності від призначення.

1.2 Принципи функціонування пристроїв надоперативної пам'яті

Як видно, з розглянутих типових архітектур побудови комп'ютера (рисунки 1.1 та 1.2) та ієрархічного принципу побудови пам'яті, інформація для процесора зберігається у внутрішній пам'яті. Вона поділяється на регістрову та кеш-пам'яті, які функціонують як пристрої з довільним доступом.

Сучасні процесори, оснащені кешем, який складається з 1-, 2- або 3-ох рівнів. Звичайно ж, бувають і винятки, але найчастіше це саме так. Загалом, можуть бути такі рівні: L1 (перший рівень), L2 (другий рівень), L3 (третій рівень). Кеш першого рівня (L1) - найбільш швидкий рівень кеш-пам'яті, який працює безпосередньо з ядром процесора, завдяки цій взаємодії, даний рівень має найменшим часом доступу і працює на частотах близьких процесору. Є буфером між процесором і кеш-пам'яттю другого рівня.

Кеш другого рівня (L2) - другий рівень більш масштабний, ніж перший, але в результаті, має менші характеристики продуктивності. Відповідно, служить буфером між рівнем L1 і L3.

Кеш третього рівня (L3) - третій рівень більш повільний, ніж два попередніх. Але все одно він набагато швидший, ніж оперативна пам'ять. Об'єм кешу L3 в i7-3770K складає 8 Мбайт. Якщо два попередніх рівня поділяються на кожне ядро, то даний рівень є загальним для всього процесора. Показник досить солідний, але в наступних поколіннях планується на рівні 20Мбайт і вище.

В даний час цифрова електроніка та схемотехніка в переважній більшості базується на використанні великих (ВІС) і надвеликих

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

інтегральних схемах (НВІС). У той же час можна сказати, що в основі ВІС і НВІС лежить інтеграція простих інтегральних схем.

Процес функціонування схем статичної пам'яті, що використовуються для побудови регістрової та кеш-пам'яті передбачає використання асинхронного інтерфейсу, що включає шину адреси, шину даних і сигнали управління CS#, OE# і WE#. Мікросхема вибирається низьким рівнем сигналу CS# (Chip Select), низький рівень сигналу OE# (Output Enable) відкриває вихідні буфери для зчитування даних, низьким рівнем WE# (Write Enable) дозволяється запис. Часові діаграми циклів звернення наведені на рисунку 1.5.

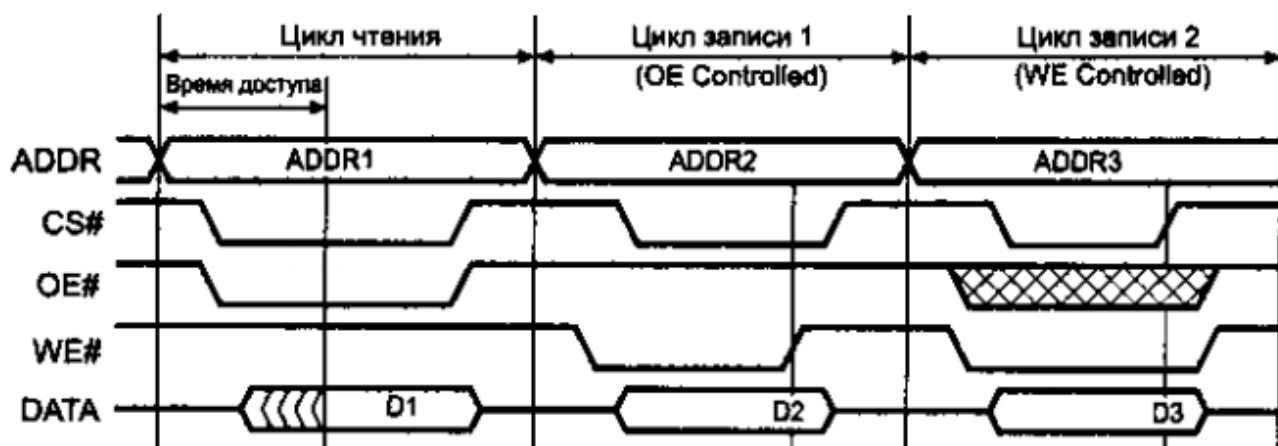


Рисунок 1.5 - Часові діаграми циклів звернення до статичної пам'яті

При запису інформації управління вихідними буферами може проводитися як сигналом OE# (цикл 1), так і сигналом WE# (цикл 2). Для зручності об'єднання мікросхем внутрішній сигнал CS# може збиратися за схемою «І» з декількох зовнішніх, наприклад CS0#, CS1# і CS2#. В такому випадку мікросхема вибирається поєднанням логічних сигналів 0, 1, 0 на відповідних входах.

Існують також пристрої статичної пам'яті, які базуються на більш складному синхронному інтерфейсі обміну інформацією. Синхронна пакетна

статична пам'ять (Sync Burst SRAM) оптимізована під виконання пакетних (burst) операцій обміну, властивих кеш-пам'яті. В її структуру введено внутрішній лічильник адреси. На додаток до сигналів, характерних для асинхронної пам'яті (адреса, дані, CS#, OE# і WE#), синхронна пам'ять використовує сигнал CLC (Clock) для синхронізації з системною шиною і сигнали управління пакетним циклом ADSP#, CADS# і ADV#. Сигнали CADS# (Cache Address Strobe) і ADSP# (ADdress Status of Processor), якими процесор або кеш-контролер зазначає фазу адреси чергового циклу, є стробами запису початкової адреси циклу у внутрішній регістр адреси. Будь-який з цих сигналів ініціює цикл звернення, одиночний (single) або пакетний (burst), а сигнал ADV# (ADVance) використовується для переходу до наступного адресою пакетного циклу. Всі сигнали, крім сигналу управління вихідними буферами OE#, синхронізуються по позитивному перепаду сигналу CLK. Це означає, що значення вхідних сигналів повинно встановитися до перепаду і утримуватися після нього ще деякий час. Вихідні дані при зчитуванні у час цього перепаду залишаються дійсними. На рисунку 1.6 наведені діаграми декількох варіантів циклів читання синхронної статичної пам'яті [].

Звернемо увагу на те що, що лічильник адреси не дозволяє перейти межу чотирьохелементного пакетного циклу. Крім того, порядок відліку адрес всередині пакетного циклу відповідає специфічному порядку (interleaved), прийнятому в процесорах i486 і вище. Мікросхеми синхронної статичної пам'яті, як і SDRAM, зазвичай мають сигнал, що вибирає режим зчитування адреси: чергування (для процесорів Intel) або послідовний рахунок (для Power PC). Синхронний інтерфейс з таким набором сигналів дозволяє пам'яті дізнаватися про наміри процесора раніше і при затримці даних на виході SRAM щодо синхронізуючого перепаду ТКQ (Clock-to-Output Access Time) 8,5, 10 і 13,5 нс забезпечувати цикл 2-1-1 -1 на частотах 66, 60 і 50 МГц відповідно. Однак на частотах 75 МГц і вище виходить цикл 3-2-2-2.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

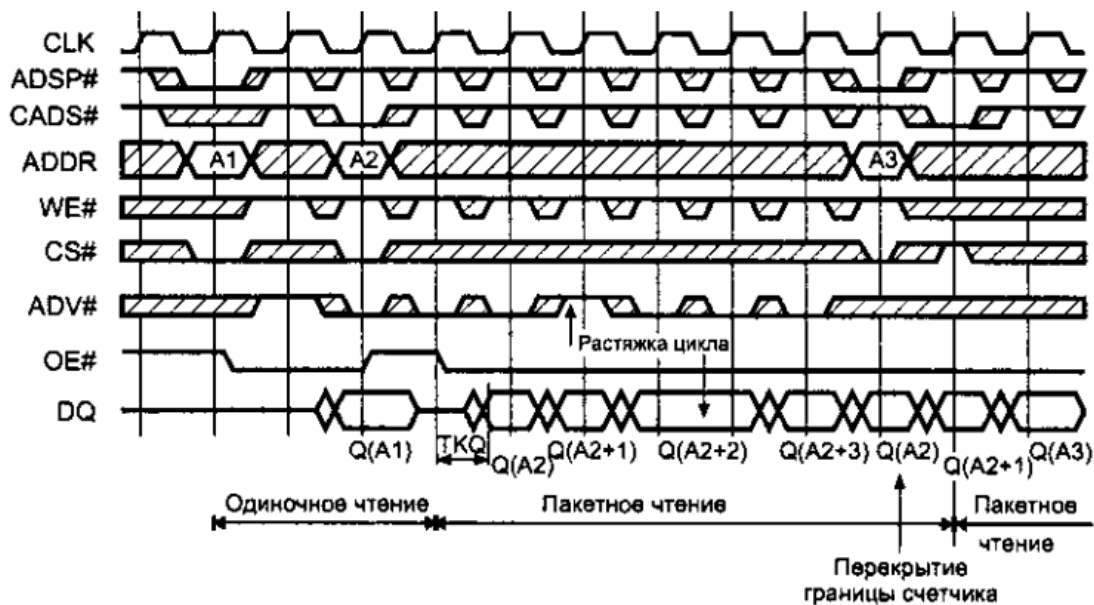


Рисунок 1.6 – Часові діаграми синхронного інтерфейсу роботи з статичною пам'яттю

Існують і інші підходи до побудови статичної пам'яті, зокрема, конвеєрно-пакетна статична пам'ять (Pipelined Burst SRAM, PB SRAM), яка є удосконаленням синхронної пам'яті.

Таким чином, в даному розділі розглянуто принципи побудови та функціонування статичної пам'яті з синхронним та асинхронним режимами роботи.

1.3 Постановка задачі по реалізації пристрою

Реалізація пристрою надоперативної пам'яті на основі архітектури кластерного розподіленого регістрового файлу передбачатиме виконання робіт по двом крупним стадіям. Перша стадія передбачає виконання проектних робіт, а друга стадія розробку електричних стем, які реалізують кластерний підхід до побудови надоперативної пам'яті.

В проектній частині передбачається деталізацію процесу обробки інформації та структури апаратної реалізації кластерного розподіленого реєстрового файлу. На основі деталізованої структури пропонується розробити інтерфейс обміну інформацією між реєстровим файлом та функціональними вузлами процесора, схему контролера управління інтерфейсом.

З метою практичної реалізації електричної схеми пристрою надоперативної пам'яті на основі архітектури кластерного розподіленого реєстрового файлу та необхідності забезпечити автоматизацію процесу розробки, слід обґрунтувати та вибрати систему автоматизованого проектування. При її виборі основним критерієм буде наявність в її базі компонентів SPICE – моделей необхідних електричних елементів та мікросхем. Важливим компонентом є розробка методики налаштування пристрою пам'яті.

В третьому розділі пропонується здійснити розробку схеми банку даних, тобто схеми набору мікросхем пам'яті. Окремий розділ слід присвятити розробці схеми контролера пристрою пам'яті. В третьому розділі планується здійснити налаштування схеми пристрою у відповідності з методикою налаштування.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ КОМПОНЕНТІВ ПРИСТРОЮ

2.1 Методики побудови пристроїв реєстрової пам'яті

Реєстровий файл (надоперативна пам'ять) - це набір програмно-доступних реєстрів, які знаходяться в реєстровій пам'яті процесора. В сучасних процесорах реєстровий файл займає одне з центральних місць. Він використовується для локального збереження операндів, адрес команд та даних, індексів, а також дозволяє організовувати ефективний обмін даними між операційними пристроями процесора та основною пам'яттю. Вибір ефективної організації реєстрового файла є одним із підходів, що дозволяє підвищити продуктивність комп'ютера. Як правило, організація реєстрового файла належить до технічних характеристик комп'ютера. Розглянемо деякі історичні аспекти розвитку організації реєстрового файла процесора. Програмно-доступні реєстри почали використовуватися з початку 1960-х років. В 1964 році фірма ІВ М розробила серію універсальних комп'ютерів ІВМ/360 для наукових та комерційних розрахунків, в процесорах яких був використаний реєстровий файл, що включав 16 32-розрядних цілочисельних реєстрів та 16 64-розрядних реєстрів з рухомою комою. Також в 1964 році був виготовлений перший суперкомп'ютер для наукових розрахунків CD C 6600, процесор якого мав реєстровий файл, що включав 24 реєстри. В 1977 році був спроектований перший векторний суперкомп'ютер Сгау-1, реєстровий файл якого мав ієрархічну структуру. Його було поділено на файл основних реєстрів і файл другорядних (фонових) реєстрів. Кількість основних реєстрів була меншою, що дозволяло забезпечити швидкий доступ до них, тоді як кількість другорядних реєстрів була більшою, проте доступ до них був повільнішим. Операнди тривалого зберігання розміщувалися в другорядних реєстрах і переміщувалися в основні реєстри при потребі їх використання. Сгау-1 загалом містив 656 реєстрів даних і адрес (включаючи векторні реєстри, але не враховуючи реєстрів керування). На особливу

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

увагу заслуговує група з восьми векторних реєстрів. Кожний такий реєстр міг вміщувати 64-х елементний вектор з рухомою комою. Однією 16-ти розрядною командою можна було додати, відняти чи перемножити два вектори. З наведених прикладів видно, що для організації високопродуктивних обчислень важливо не тільки мати велику кількість реєстрів, але і забезпечити швидкий доступ до даних, які в них зберігаються, за рахунок ефективної організації реєстрового файлу. На рисунку 2.1 приведена класифікація типів реєстрових файлів процесора. В них використовується статична організація збереження даних, тобто при записі і читанні даних вказується безпосередня адреса реєстра, яка є незмінною під час виконання програми.

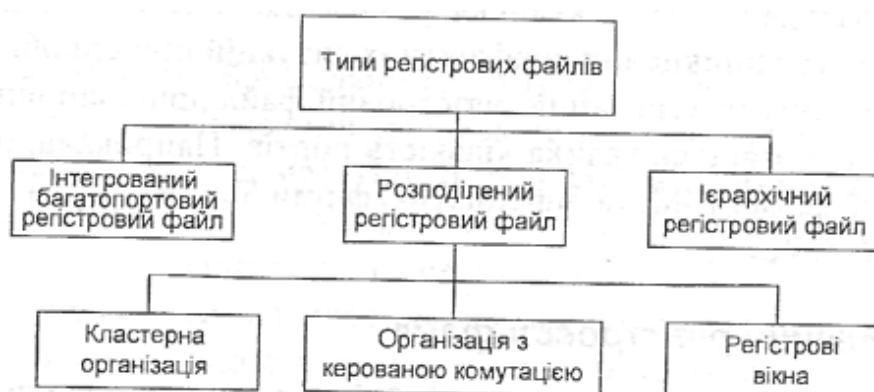


Рисунок 2.1 - Класифікація типів реєстрових файлів

В інтегрованому багатопортовому реєстровому файлі забезпечується доступ до будь-якого реєстру з кожного його порту. Вихід кожного реєстра під'єднаний до відповідних інформаційних входів мультиплексорів. Адреси, що подаються на входи керування мультиплексорів, визначають номери реєстрів, з яких читаються дані. Для запису використовується демультимплексор ДМП, через який на відповідний реєстр подається сигнал запису даних, що поступає з вхідної шини, та адреси реєстра. Процес запису

синхронізується тактовими імпульсами. Кожний з регістрів має вхід дозволу запису даних, інформаційний вхід, та інформаційний вихід.

Недоліком даної архітектури є те, що із збільшенням кількості портів зчитування та запису збільшується кількість мультиплексорів та демультиплексорів, а також суттєво збільшується кількість шин передачі даних, що ускладнює реалізацію регістрового файлу. Крім того, з'являється можливість виникнення конфліктних ситуацій при спробі запису даних до тих самих регістрів. Тому інтегрований регістровий файл доцільно використовувати в комп'ютерах, де не вимагається велика кількість портів.

Використання розподілених регістрових файлів дозволяє зменшити площу кристалу, яку займає регістрова пам'ять процесора. Розподілення регістрового файлу здійснюється шляхом поділу функціональних елементів процесора на групи, кожна з яких має свій локальний регістровий файл. Відповідно, такі регістрові файли містять меншу кількість портів та регістрів, що зменшує затрати обладнання на їх реалізацію. Існує три типи розподілених регістрових файлів: кластерні, з керованою комутацією та з віконною організацією.

Локальні регістрові файли можуть бути повністю незалежними, тобто дані для певної групи функціональних елементів процесора доступні тільки з конкретного локального регістрового файлу. Такий розподілений регістровий файл, який складається з незалежних локальних регістрових файлів, відповідно до наведеної вище класифікації, називають кластерним. Кластерні регістрові файли здебільшого використовуються у векторних процесорах. В таких процесорах одна операція над всіма n компонентами векторних операндів задається однією командою. Кожен з таких векторних операндів міститься в локальному регістровому файлі біля відповідного функціонального елемента процесора, і відпадає необхідність у повноцінних зв'язках між локальними регістровими файлами. На рисунку 2.2 показана кластерна організація регістрового файлу процесора TMS320 C64x фірми Texas Instruments.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

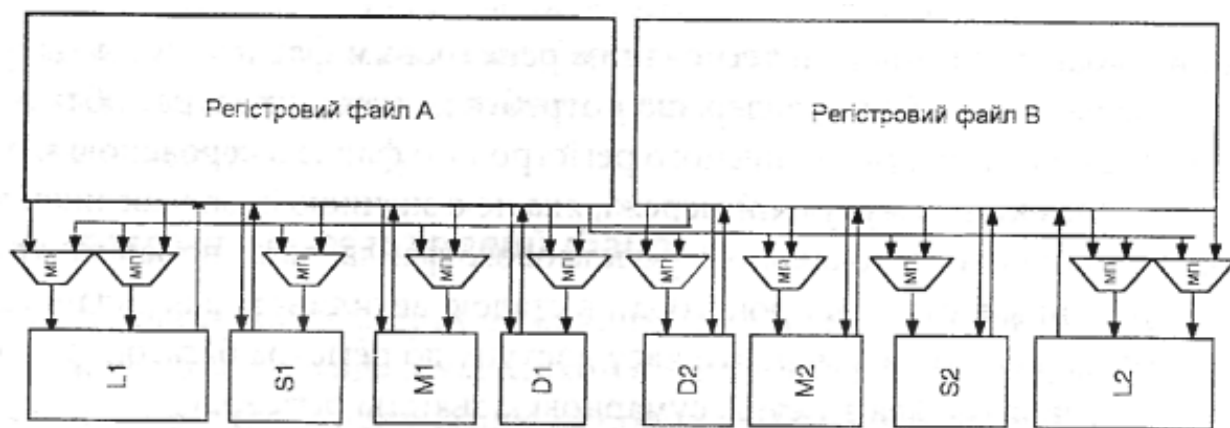


Рисунок 2.2- Організація кластерного реєстрового файла

В цьому процесорі функціональні елементи поділені на підмножину А (L1, S1, M1, D1) та підмножину В (L2, S2, M2, D2). Функціональні елементи процесора L (L1, L2) виконують арифметичні операції та операції порівняння. Функціональні елементи процесора S (S1, S2) виконують арифметично-логічні операції та команди керування. Функціональні елементи процесора M (M1, M2) виконують множення 16-ти розрядних операндів, а функціональні елементи процесора D (D1, D2) виконують арифметичні операції, та виконують роль генераторів адрес.

Розглянута структура служить основою для її деталізації в наступному розділі.

2.2 Проектування структури модуля реєстрової пам'яті на основі кластерного підходу

Як було показано в розділі 2.1 ідея кластерного розподіленого реєстрового файлу полягає в тому, що для функціонування конкретної групи функціональних вузлів процесора створюється свій незалежний реєстровий файл. Тобто, створюється структура, що реалізує апаратний паралелізм в

обробці інформації. Даний підхід використовується у векторних процесорах. Векторний процесор - це процесор, в якому операндами деяких команд можуть виступати впорядковані масиви даних - вектори. Відрізняється від скалярних процесорів тим, що ті можуть працювати тільки з одним операндом в одиницю часу. Векторні процесори поширені в сфері наукових обчислень, де вони є основою більшості. Однак різке збільшення продуктивності і активна розробка нових продуктивних скалярних процесорів привели до витіснення векторних процесорів зі сфери повсякденних процесорів. У більшості сучасних мікропроцесорів є векторні розширення (SSE). Крім того, сучасні відеокарти і фізичні прискорювачі можна розглядати як векторні співпроцесори.

Архітектура регістрового файлу передбачає наявність в його структурі мультиплексорів, демультимплексорів, регістрів – рисунок 2.3. В інтегрованому багатопортовому регістровому файлі забезпечується доступ до будь-якого регістру з кожного його порту.

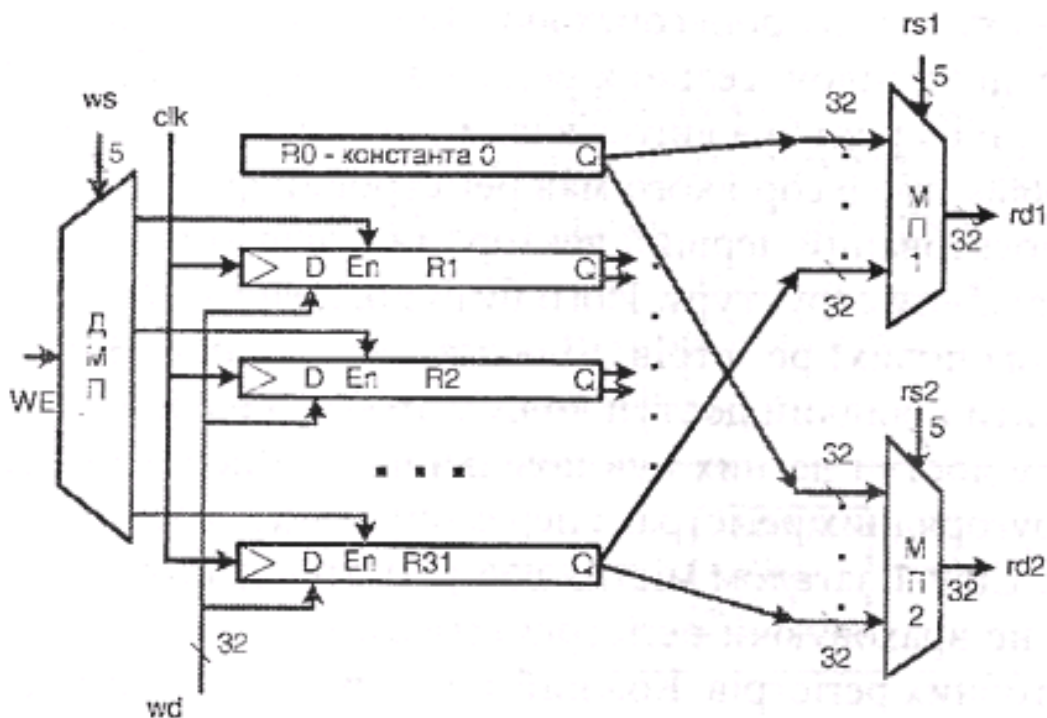


Рисунок 2.3 – Структура регістрового файлу

На рисунку показана структура інтегрованого реєстрового файлу з двома портами зчитування (rd1, rd2), та одним портом запису (wd), який містить 32 32-розрядних реєстри (R0-R31). В реєстр R0 запис заборонений, оскільки в ньому зберігається константа нуль. Вихід кожного реєстра під'єднаний до відповідних інформаційних входів мультиплексорів МП1 та МП2. Адреси (rs1, rs2), що подаються на входи керування мультиплексорів, визначають номери реєстрів, з яких читаються дані. Для запису використовується демультимплексор ДМП, через який на відповідний реєстр подається сигнал запису даних (we), що поступає з вхідної шини, та адреса реєстра ws. Процес запису синхронізується тактовими імпульсами, що подаються на вхід clk. Кожний з реєстрів має вхід дозволу запису даних Ep, інформаційний вхід D, та інформаційний вихід Q.

З врахуванням архітектур реєстрового файлу та кластерного розподіленого реєстрового файлу розробимо його деталізовану структурну схему – рисунок 2.4.

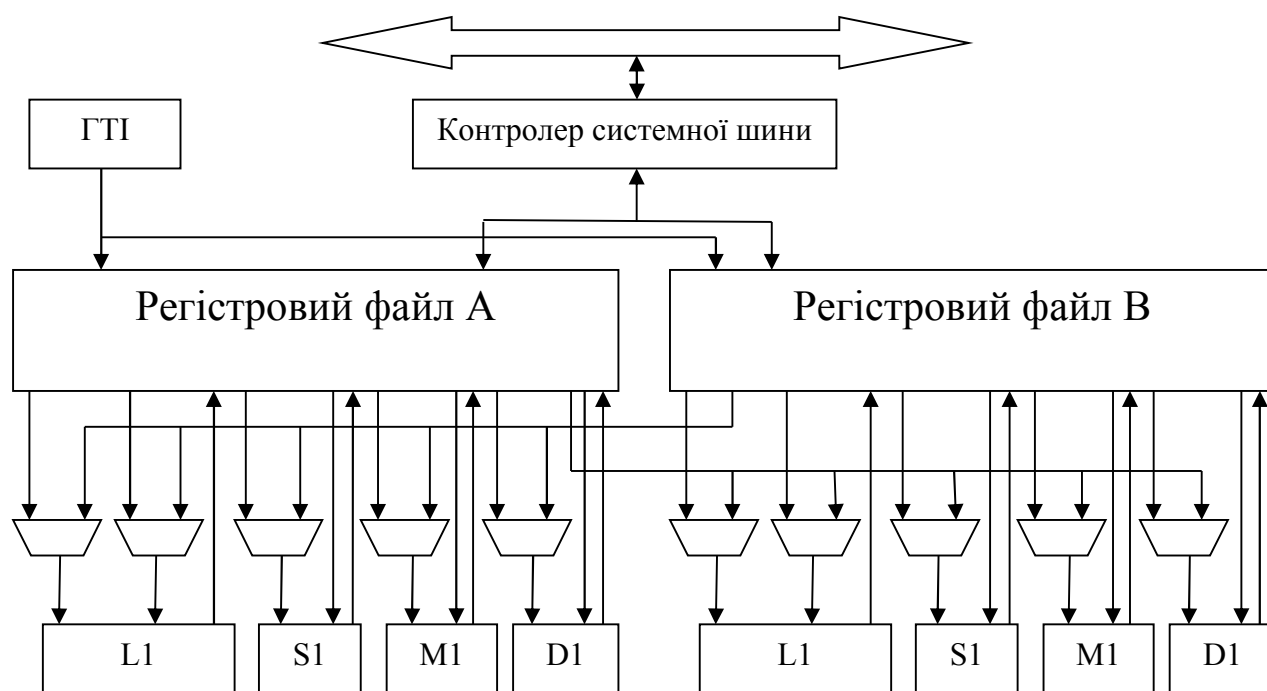


Рисунок 2.4 – Узагальнена структурна схема розподіленого кластерного реєстрового файлу

При розробці узагальненої структурної схеми розподіленого кластерного реєстрового файлу були враховані матеріали, що описують схеми 2.2 та 2.3 і доповнені пристроями генерації тактових імпульсів (ГТІ) та блоком контролера системної шини. Його призначення розшифровувати команди, які поступають по системній шині (запис/читання) та приймати чи передавати дані в локальні реєстрові файли.

Разом з тим аналіз процесу вводу/виводу інформації на функціональні вузли L_i , S_i , V_i , D_i потребує уточнення та введення в структуру на рисунку 2.4 схеми управління (контролера) цим процесом. З врахуванням цих уточнень деталізована структурна схема розподіленого кластерного реєстрового файлу набуде наступного виду – рисунок 2.5.

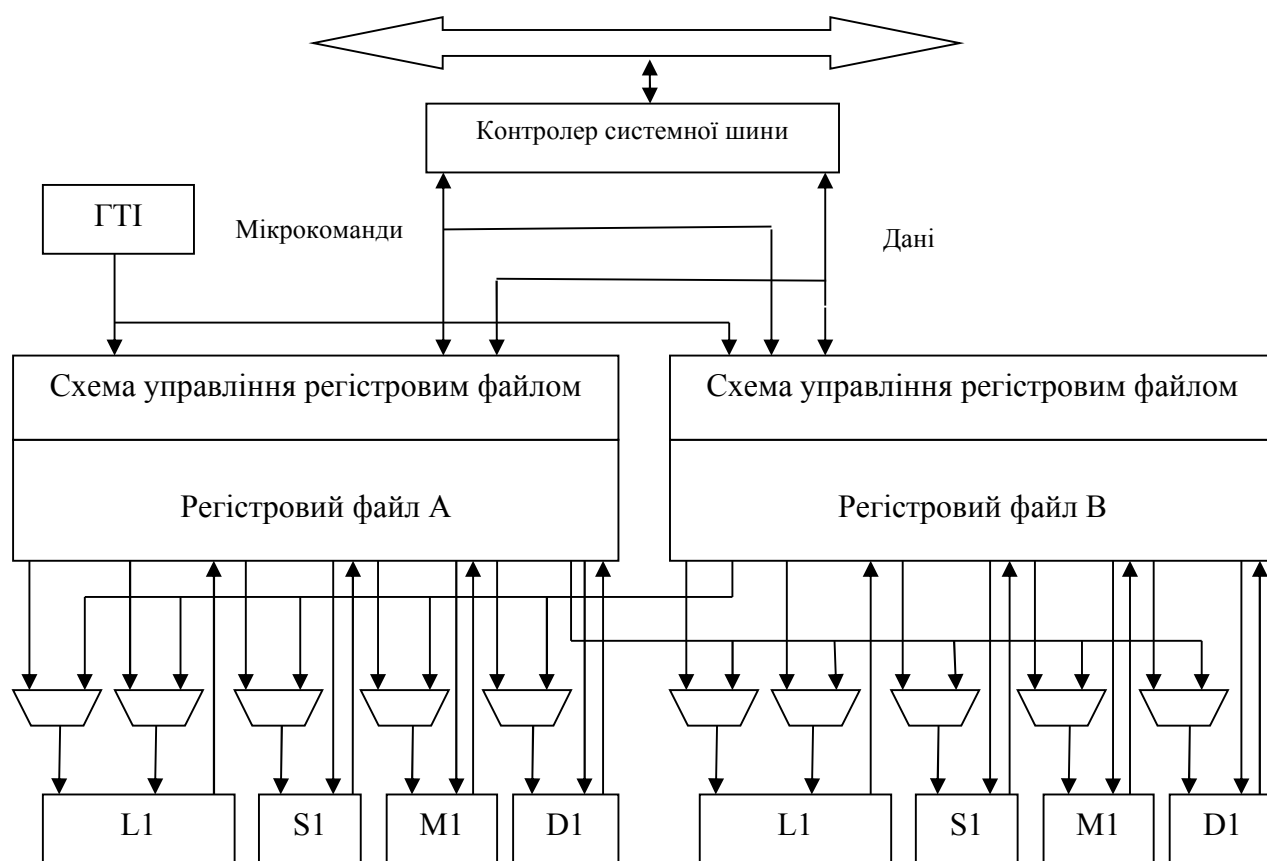


Рисунок 2.4 – Деталізована структурна схема розподіленого кластерного реєстрового файлу

Як видно з розробленої структури мікрокоманда управління процесом обробки інформації повинна містити на ступні складові:

- код кластерного реєстрового файлу;
- код реєстру (комірки пам'яті);
- команду запису/зчитування;
- код векторного процесора;
- код функціонального вузла векторного процесора.

Це означає, що доступ до цих функціональних вузлів відбувається шляхом задання їх адреси. основним елементами контролера будуть дешифратори. Комутуючими елементами будуть тривходові мультиплектори.

Таким чином, розроблена узагальнена структурна схема функціонування розподіленого кластерного реєстрового файлу повинна стати основою для наступної її апаратної реалізації.

2.3 Структура контролера управління кластерним реєстровим файлом

Розробку структури контролера управління слід розпочати з деталізації процесу обміну інформацією та команд, що при цьому використовуються. Вважаємо наявні два основних процеси роботи кластерного файлу – зчитування та запис. На етапі зчитування пропонується визначати кластер, який буде видавати інформацію, реєстри (два) в комірках яких вона знаходиться, процесор та вузол, які оброблятимуть інформацію.

При цьому необхідно визначитися зі структурою зв'язку між реєстровим файлом та функціональними вузлами векторного процесора. В даному випадку, враховуючи, що на вхід функціональних вузлів поступають два веранди. Можна піти двома шляхами – ввести шину даних 1 та шину даних 2, а можна використати мультиплексування однієї шини. Враховуючи,

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

що перший варіант більш швидкодіючий (використовує один такт) приймаємо його за базовий. Схема зв'язку кластерного регістрового файлу та функціональних вузлів процесора приведена на рисунку 2.5.

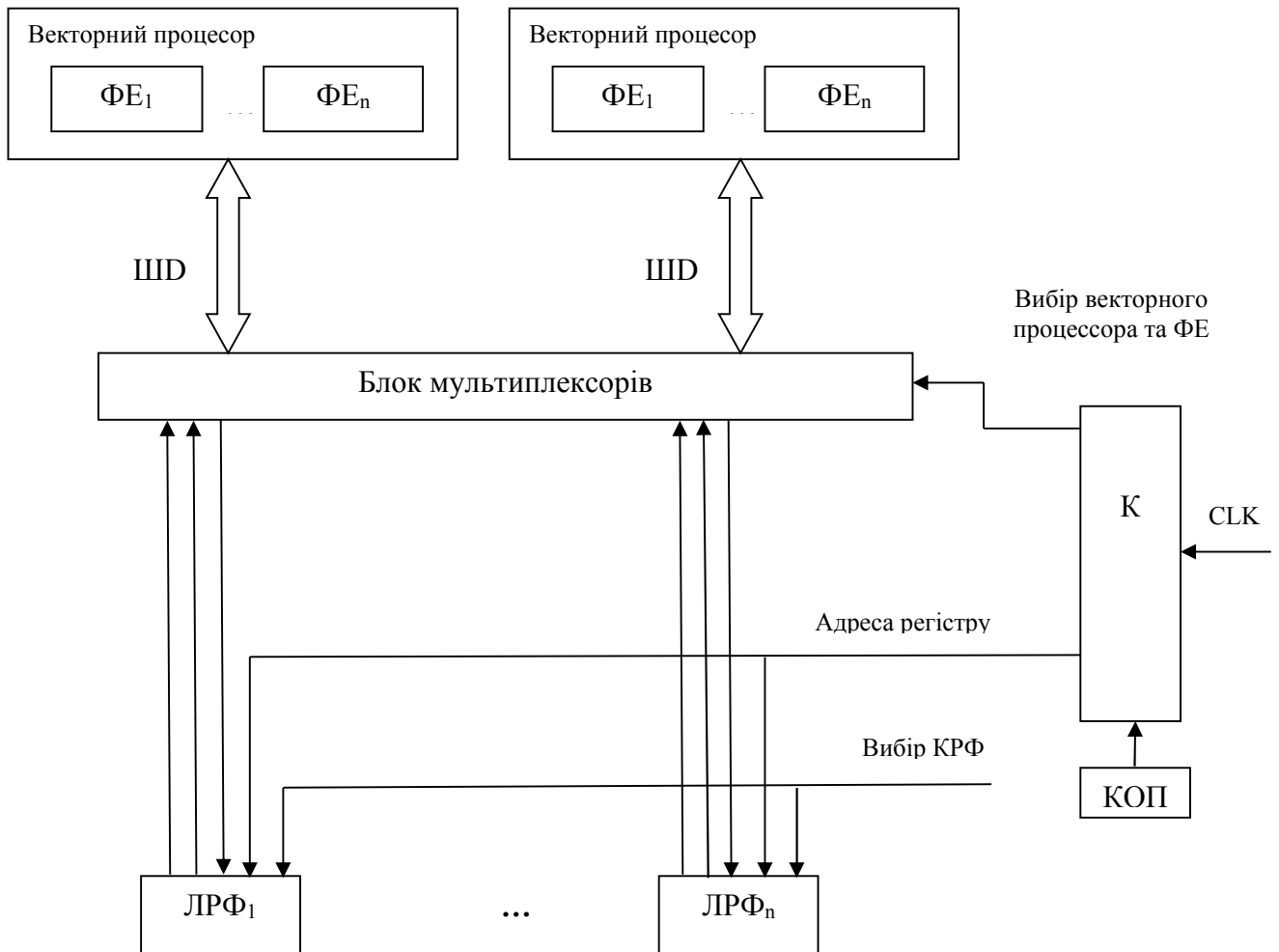


Рисунок 2.5 - Схема зв'язку кластерного регістрового файлу та функціональних вузлів процесора

Дана схема служить основою для проектування схеми контролера управління кластерним регістровим файлом. Як було сказано в розділі 2.2 та уточнено в даному розділі, основними компонентами контролера є дешифратор адреси ФЕ, дешифратор адреси комірки та дешифратор процесу зчитування/запис. Структурна схема приведена на рисунку 2.6.

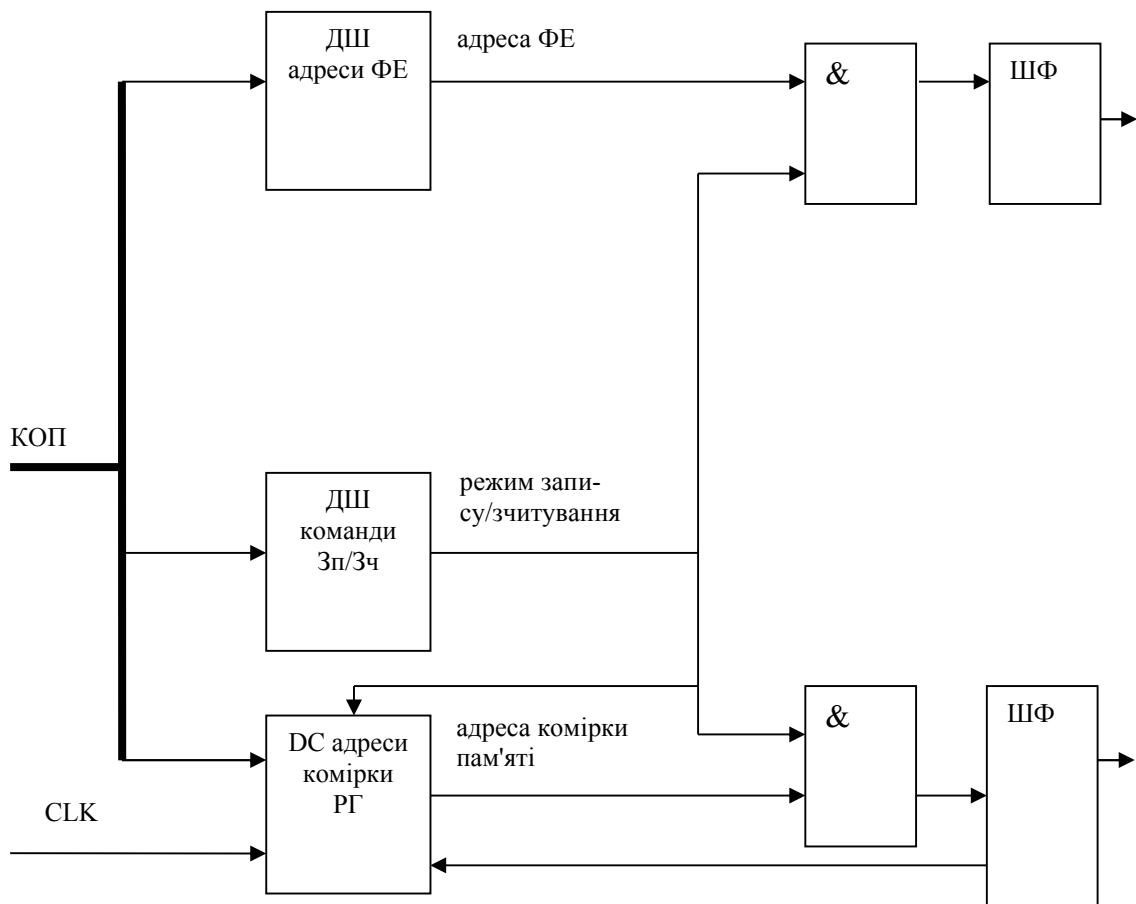


Рисунок 2.6 – Схема управління процесом обміну інформацією

Таким чином, запропонована схема служить основою для апаратної реалізації контролера управління кластерним регістровим файлом.

2.4 Обґрунтування вибору системи автоматизованого проектування пристрою

Для автоматизації процесу розробки та верифікації електричних схем цифрових пристроїв використовуються системи імітаційного моделювання. В даному розділі розглянемо основні можливості та методи роботи системи схемотехнічного моделювання NI Multisim. NI Multisim, що призначена для

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						32
Зм.	Арк.	№ докум.	Підпис	Дата		

моделювання та аналізу електричних схем. Програма NI Multisim дозволяє моделювати аналогові, цифрові та цифро-аналогові схеми великий ступеня складності. Наявні в програмі бібліотеки включають в себе великий набір широко поширених електронних компонентів. У програмі передбачена можливість підключення та створення нових бібліотек компонентів. Параметри компонентів можна змінювати в широкому діапазоні значень. Моделі цих компонентів згруповані у відповідні групи, що розміщуються у панелі моделей електричних компонентів. Прості компоненти описуються набором параметрів, значення яких можна змінювати безпосередньо з клавіатури, активні елементи - моделлю, яка представляє собою сукупність параметрів і описує конкретний елемент або його ідеальне уявлення.

Модель конкретного елемента вибирається з відповідної панелі бібліотеки компонентів (рисунок 2.7).

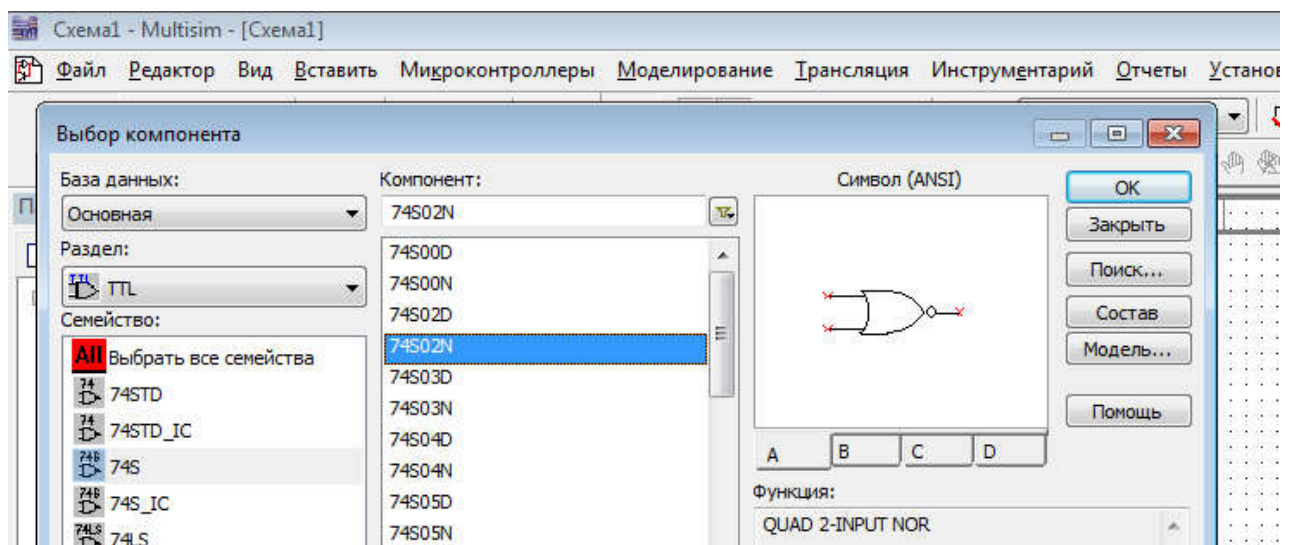


Рисунок 2.7 – Вибір моделі електричного елемента з відповідної моделі.

Широкий набір приладів дозволяє проводити вимірювання різних величин, задавати вхідні дії, будувати графіки. Всі прилади зображуються у вигляді, максимально наближеному до реального, тому працювати з ними

просто і зручно. Результати моделювання можна вивести на принтер або імпортувати в текстовий або графічний редактор для їх подальшої обробки.

Основні переваги даної програми є:

- Економія часу. Робота в реальній лабораторії вимагає великих часових витрат на розробку схемної реалізації та підготовку експерименту.
- Достовірність вимірювань. В NI Multisim всі елементи описуються строго встановленими параметрами, тому кожен раз в ході експерименту буде повторюватися результат, який визначається тільки параметрами елементів і алгоритмом розрахунку.
- Зручність проведення вимірювань. Працюючи з NI Multisim, виконавець застрахований від випадкового ураження струмом, а прилади не вийдуть з ладу у випадку невірно зібраної схеми.
- Графічні можливості. NI Multisim дозволяє розмістити схему таким чином, щоб було чітко видно всі з'єднання елементів і одночасно вся схема цілком. Можливість зміни кольору провідників дозволяє зробити схему більш зручною для сприйняття.
- Стандартний інтерфейс Windows.
- Сумісність з програмою P-SPICE. Програма NI Multisim базується на стандартних елементах програми SPICE. Це дозволяє експортувати різні моделі елементів і проводити обробку результатів, використовуючи додаткові можливості різних версій програми P-SPICE.

Після розробки електричної принципової схеми цифрового пристрою в NI Multisim розробник має можливість транслювати її в САПР Ultiboard для розробки друкованої плати.

Вказані особливості NI Multisim обумовлюють її використання.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

3 АПАРАТНА РЕАЛІЗАЦІЯ ТА ВЕРИФІКАЦІЯ ПРИСТРОЮ НАДОПЕРАТИВНОЇ ПАМ'ЯТІ .

3.1 Розробка схеми контролера інтерфейсу та його налаштування

На першому етапі апаратної реалізації контролера кластерного регістрового файлу розробимо схему блоку мультиплексорів. Цифровий мультиплексор, або селектор даних, є логічною комбінаційною схемою, яка приймає кілька двійкових сигналів і вибирає один і них і передає на вихід. Передача необхідного сигналу на вихід визначається входом вибору даних (SELECT), який іноді називають входом вибору адреси (ADDRESS). На рисунку 3.1 представлено умовно-графічні позначення різних мікросхем мультиплексорів.

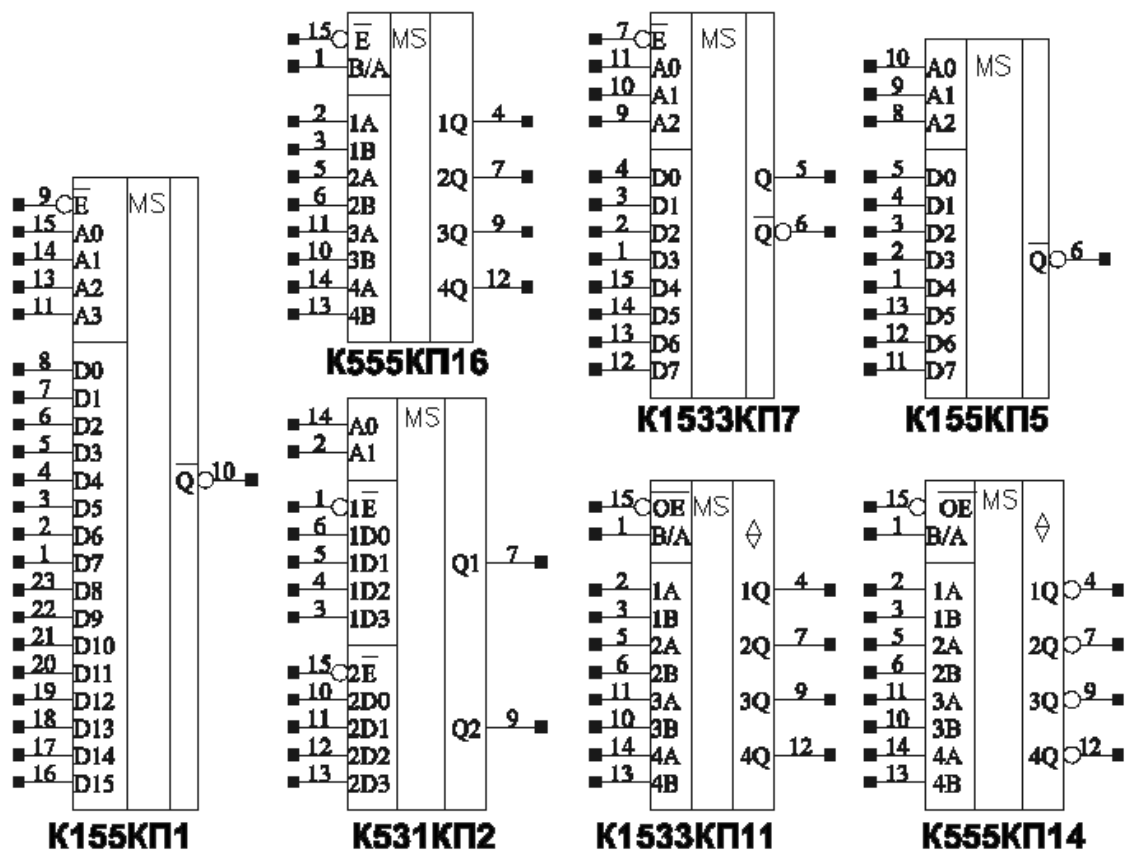


Рисунок 3.1 – Умовно-графічні позначення мультиплексорів

Для вибору мультиплексоре потрібне деяке уточнення. Будемо вважати, що векторний процесор містить 4 арифметико-логічних пристрої. Це означає, що мультиплексор який формує дані певного кластерного регістрового файлу, повинен мати організацію 2^x1 . Однак в структурі кластерного розподіленого файлу (рисунок 2.4) передбачено, що кожен кластер має один порт для зчитування даних р інших КРФ. Тому організація мультиплексорів в режимі зчитування повинна мати наступну структуру – рисунок 3.2.

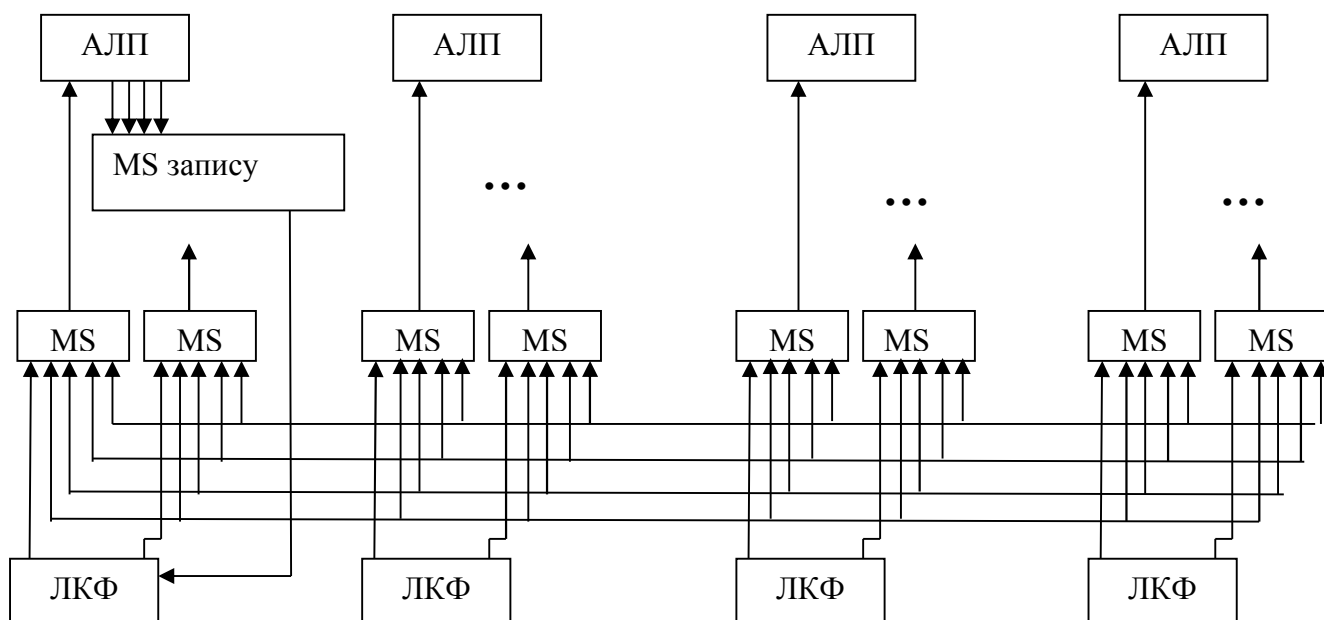


Рисунок 3.2 – Організація блоку мультиплексорів

З врахуванням даних рисунку 3.1 та структури блоку вибираємо в якості мультиплексора MS мікросхему K1533КП7 (аналог 74НС151N). Схема дослідження мікросхеми 74НС151N побудована на основі таблиці 3.1 приведена на рисунку 3.2. Як видно з рисунку мікросхема має вісім входів, які комутуються на прямий та інверсний виходи. Як видно з рисунку 3.3, лог.0 подається на третій вхід мультиплексор U22, а сигнали на адресних

входах – ABC (010-це третій стан) забезпечить його комутацію на вихід, при подачі лог.0 на керуючий вхід.

Таблиця 3.1 – Таблиця істинності 74ALS151N

INPUTS				OUTPUTS	
S2	S1	S0	\bar{E}	Y	\bar{Y}
X	X	X	H	L	H
L	L	L	L	I0	$\bar{I}0$
L	L	H	L	I1	$\bar{I}1$
L	H	L	L	I2	$\bar{I}2$
L	H	H	L	I3	$\bar{I}3$
H	L	L	L	I4	$\bar{I}4$
H	L	H	L	I5	$\bar{I}5$
H	H	L	L	I6	$\bar{I}6$
H	H	H	L	I7	$\bar{I}7$

H = High voltage level
L = Low voltage level
X = Don't care

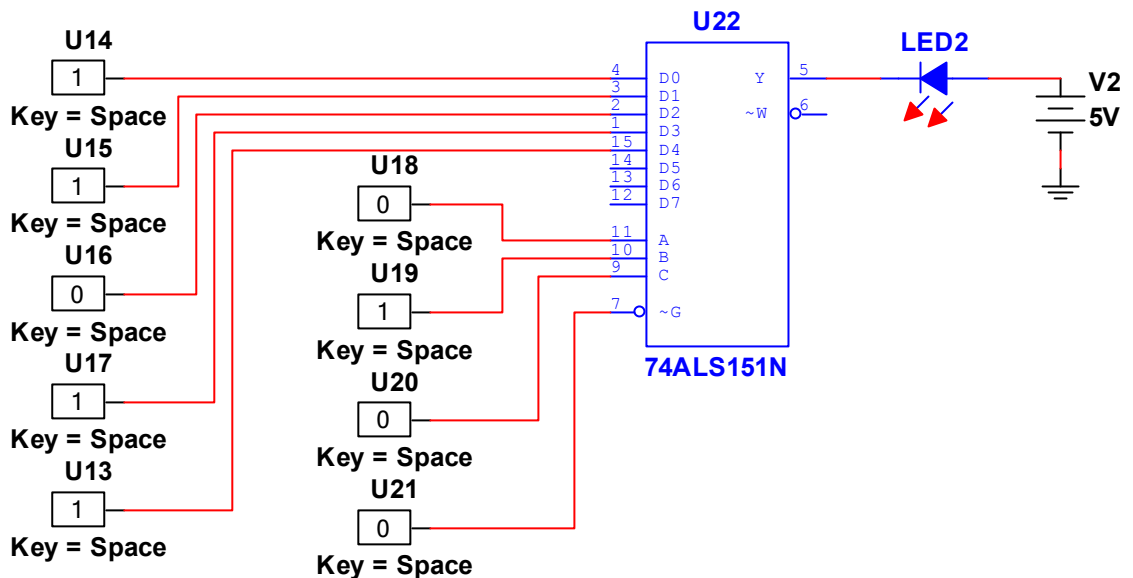


Рисунок 3.3 – Схема включення 74HC151N

Світіння світлодіоду LED2 індикуює стан виходу мультиплексор – Y. Враховуючи, що архітектурою кластерного файлу передбачено передачу

даних між кластерами (рисунок), то при розробці блоку мультиплексорів для кожного кластера створюємо пару мультиплексорів U1-U3, U2-U4, U5-U7, U6-U8. При цьому вважаємо, що перший та другий операнди подаються на перші входи пари мультиплексорів даного кластерного розподіленого файлу. На інші входи поступають дані з інших кластерів.

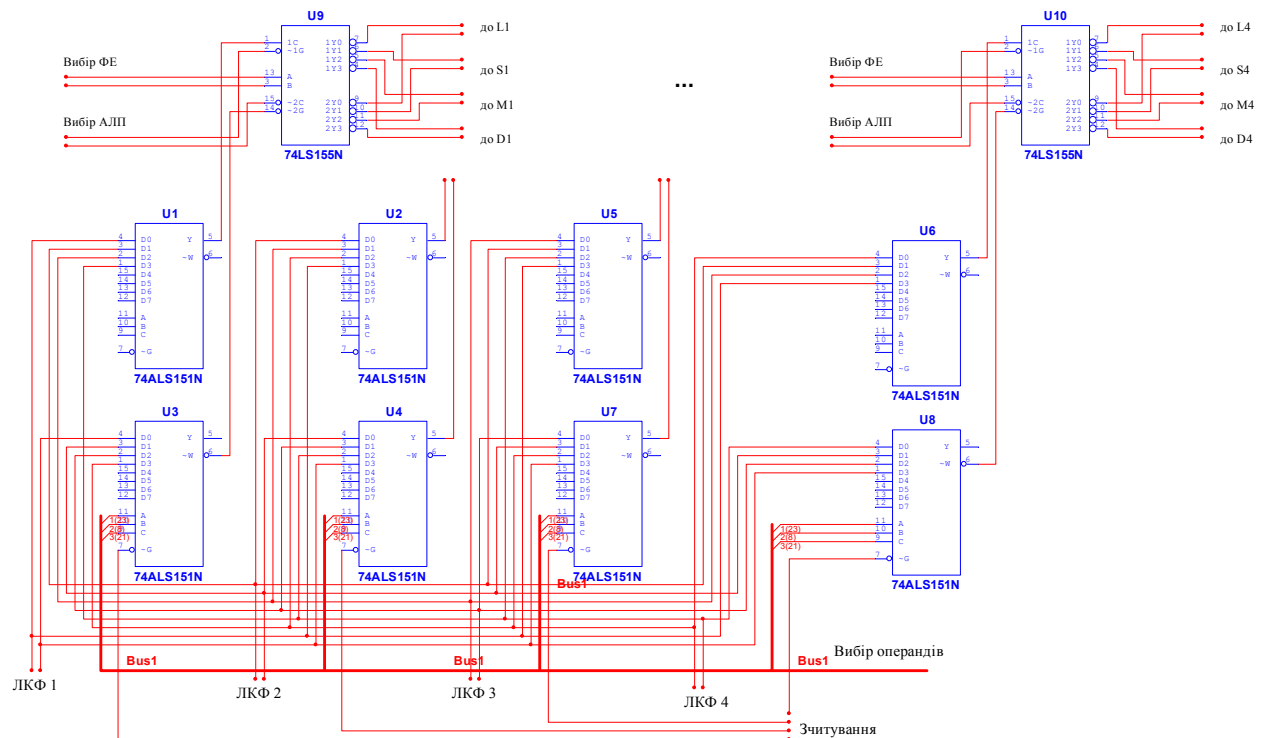


Рисунок 3.4 – Електрична схема блоку мультиплексорів

Сигнали на виході мультиплексорів визначаються кодом "вибір операндів" та сигналом "зчитування". Вихідні сигнали мультиплексорів U1-U3, U2-U4, U5-U7, U6-U8 підключаються відповідно до входів демультиплексорів U9...U10. Два демультиплексори умовно не показані. Завданням демультиплексорів є подача двох операндів на входи ФЕ (L, S, M, D) відповідного векторного процесора. Оскільки функціональних елементів є чотири, то для цього використаємо мікросхему 74LS155.

Мікросхема 74LS155 містить два 2-розрядних демультимплексори з загальними адресними входами. Вибір необхідного входу на обох демультимплексорах мікросхеми здійснюється подачею на загальні адресні входи А і В відповідного двійкового коду у відповідності з таблицею істинності 3.2. При цьому перший демультимплексор комутує перший операнд, а другий – другий.

Таблиця 3.2 - Таблиця істинності 74ALS155N

ENABLE	SELECT INPUT	INPUTS		OUTPUT
E	S	I ₀	I ₁	Z
H	X	X	X	L
L	H	X	L	L
L	H	X	H	H
L	L	L	X	L
L	L	H	X	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

Коли в мікросхемі 74155 (74LS155) – U9 – U10 (1 схема) на інформаційний вхід 1D (вивід 1) подається напруга високого рівня, а на вхід стробу-імпульсу 1G (вивід 2) - низького, то на виході, що відповідає стану адресних входів, встановлюється напруга низького рівня, на інших виходах - високого. Коли мікросхемі 74155 (74LS155) (2 схема) на інформаційний вхід 2D (вивід 15), так само як і на вхід строба-імпульсу 2G (вивід 14), подається напруга низького рівня, то таку напругу встановлюється на виході, відповідному станом адресних входів. Якщо в мікросхемі 74155 (74LS155) (схема 1) на вхід стробуючого імпульсу подається напруга низького рівня, на обраному виході з'являються вхідні дані в інверсному вигляді. Якщо в мікросхемі 74155 (74LS155) (схема 2) на вхід стробуючого імпульсу

подається напруга низького рівня, на обраному виході з'являються вхідні дані в неінвертуючому вигляді.

Особливістю мікросхеми є те, що обидві половини мікросхеми 74155 (74LS155) відрізняються одна від другої: схема 1 інвертує вхідні дані, а схема 2 - ні. Тому на вхід першого демультиплексора подаємо операнд з прямого виходу мультиплексора U1 (рисунок 3.4), а на другий – з інверсного виходу U3. Аналогічно під'єднуються входи інших демультиплексорів.

Другим компонентом блоку мультиплексорів, є набір мікросхем, що забезпечують подачу вихідного сигналу ФЕ векторного процесора на вхід свого кластерного розподіленого регістрового файлу – рисунок 3.5. Ця функція забезпечується використанням мультиплексорів побудованих по схемі 4x1 – 74ALS153.

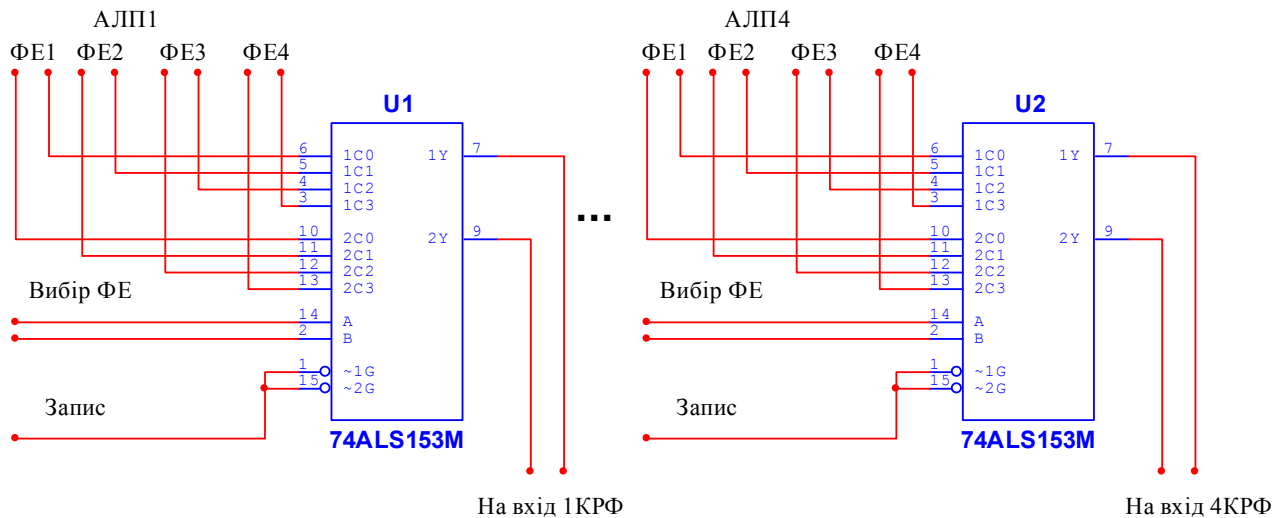


Рисунок 3.5 – Схема мультиплексорів, що забезпечують запис даних в КРФ

Схему контролера інтерфейсу, що формує сигнали управління кластерним регістровим файлом здійснюємо у відповідності зі схемою представленою на рисунку 2.6. Згідно цієї структурної схеми в схемі контролера здійснюється дешифрація трьох складових коду мікрооперацій:

коду адреси ФЕ, коду команди зчитування/запис та коду адреси КРФ та комірки (регістра) в ньому. Враховуючи, що кількість ФЕ та КРФ рівна чотирьом, то для них підходять дешифратори 2^x4 , зокрема, мікросхема 74ALS155N. Дешифратор ФЕ АЛП реалізовано в схемі на рисунку 3.4 елементи U9, U10. Тому в схемі контролера реалізуємо функцію визначення АЛП (векторного процесора) – рисунок 3.6.

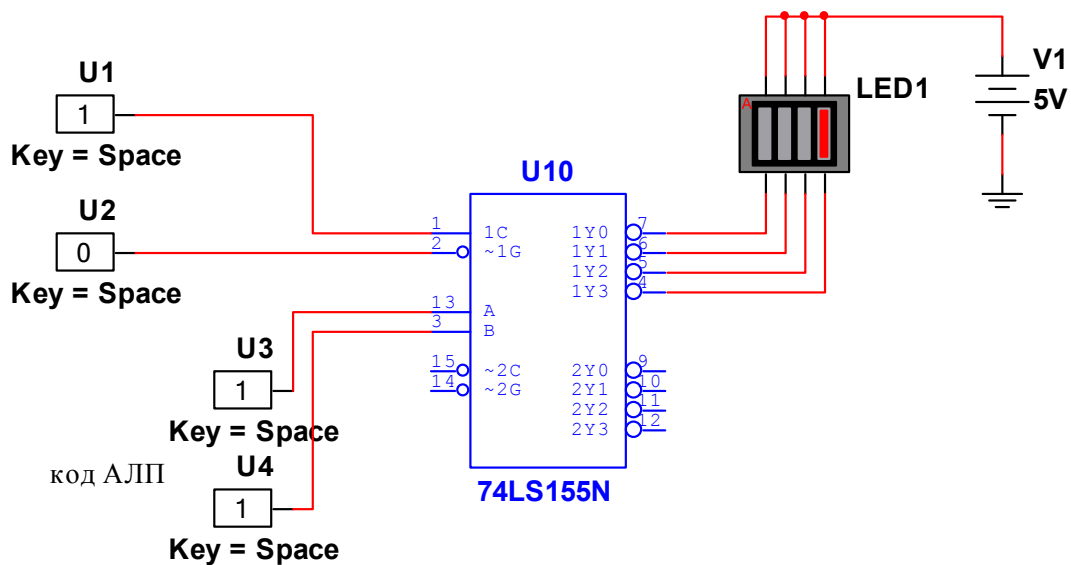


Рисунок 3.6 – Апаратна реалізація дешифратора арифметико-логічного пристрою

На схемі дешифратора арифметико-логічного пристрою код АЛП поступає на входи А і В. Вхід 1G дозволяє роботу першого дешифратора. Дешифратор команди можна побудувати з використанням інвертора та D-тригера – рисунок 3.7. Код операції запису, наприклад лог.0, по фронту CLK записується в D-тригер і використовується для організації роботи кластерного регістрового файлу.

Наступним кроком буде розробка дешифратора регістру інформації в даному кластерному регістровому файлі. враховуючи, що кількість комірок пам'яті значно більша ніж кількість кластерних регістрових файлів. Тому для

організації дешифратора регістру пам'яті слід використати пірамідальну структуру нарощення розрядності адреси регістра.

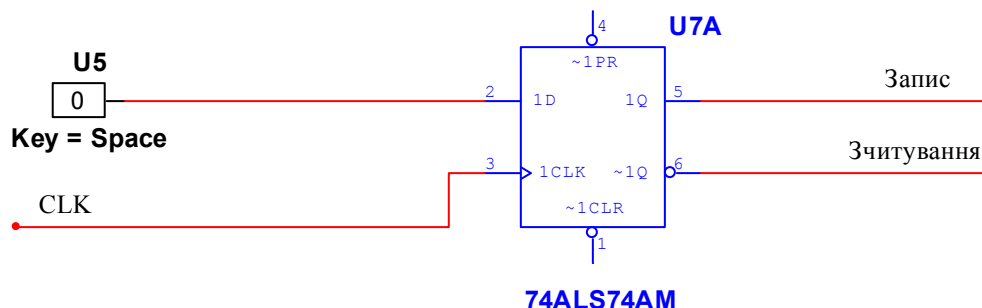


Рисунок 3.7 – Дешифратор команд регістрового файлу

Таким чином, в даному розділі апаратно реалізовано схему контролера управління процесом роботи кластерного розподіленого регістрового файлу та функціональними елементами процесора, яка послужить основою для апаратної реалізації кластера пам'яті.

3.2 Апаратна реалізація схеми кластерного розподіленого регістрового файлу

Основою для розробки КРФ служать регістри пам'яті. Регістром називають декілька тригерів, що не мають внутрішніх зворотних зв'язків. Регістри застосовуються для накопичення, зберігання та зсуву даних.

У регістрі тригери з'єднані послідовно: виходи Q попереднього тригера передають біт даних на інформаційні входи наступного. Всі тактові входи С тригерів з'єднані паралельно. При такому включенні одиниця, записана у вигляді напруг низького і високого рівнів по входах першого тригера, після подачі одного тактового імпульсу перейде в другій тригер, потім під час

наступного тактового імпульсу вона потрапить в третій тригер і так пройде далі, до кінця регістра – рисунок 3.9. Це приклад послідовного регістру, або регістру зсуву.

В регістрах також передбачена також логічну схему паралельного відображення на виходах стану кожного тригера. Тоді після заповнення регістра від послідовного або паралельних входів, за командою дозволу виходу накопичене цифрове слово можна відтворити на всіх паралельних виходах. Для зручності почергової видачі даних від таких регістрів - буферних накопичувачів в шину даних обробного пристрої - процесора - паралельні виходи регістрів забезпечуються вихідними буферними підсилювачами, мають третє, розімкнуте Z-стан. За багатопровідникової шині даних процесор отримає цифрове, слово - байт від виходів того регістра, яким дана команда дозволу видачі. Серед багатифункціональних мікросхем середнього рівня інтеграції, виконаних на логічних елементах КМОП, популярні чотирьох, восьми і дванадцятирозрядний регістри. На вибір регістрів пам'яті суттєві обмеження накладає система автоматизованого проектування NI Multisim. В її базі даних знаходяться наступні мікросхеми регістрів – рисунок 3.9. Як видно з даного рисунку перспективним для апаратної реалізації банку пам'яті кластерного реєстрового файлу будуть мікросхеми 74LS173N та 74LS373N.

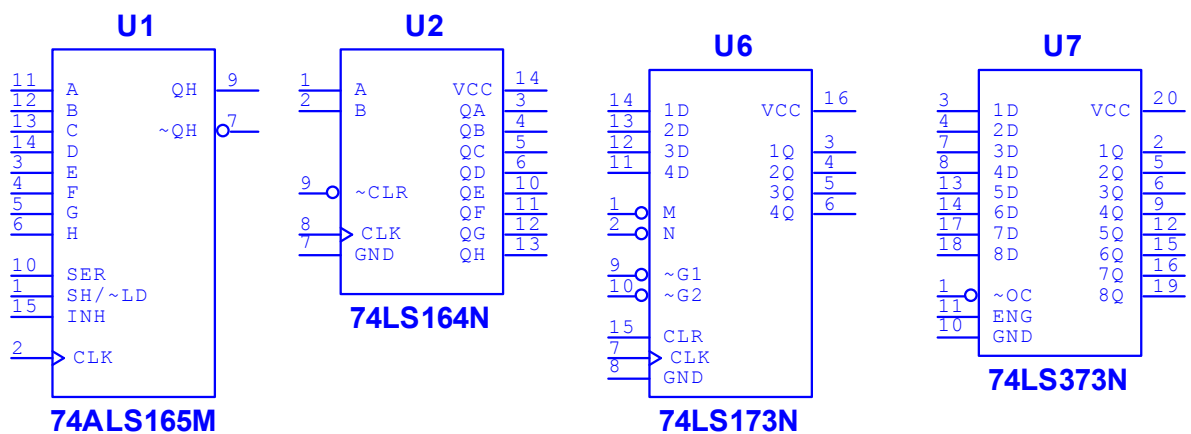


Рисунок 3.9 – Типові мікросхеми двійкових регістрів

Тому в ролі базової мікросхеми вибираємо 74LS373N. Аналіз системи функціонування регістру проведемо із використанням схеми приведеної на рисунку 3.10.

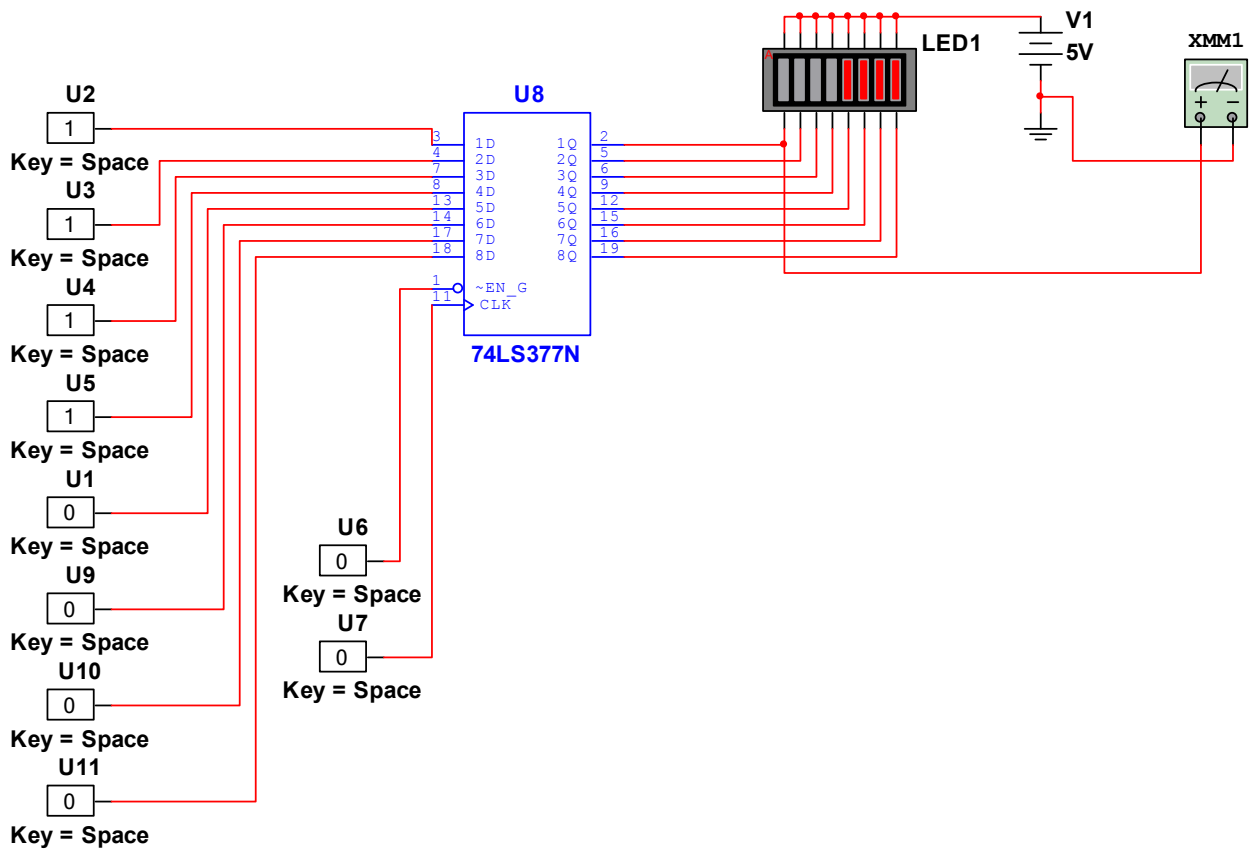


Рисунок 3.10 – Схема дослідження паралельного регістру 74LS377N

Як видно зі схеми, на вхід регістру поступає код 11110000. Під управлінням сигналу лог.0 на вході EN_G (вивід 1) та імпульсу строби CLK (вивід 11) цей код записується в регістр. Лог.0 на виході регістру викликають свічення розрядів індикатора LED1.

Вибір регістру та аналіз його роботи послужили основою для детального проектування регістрового файлу – рисунок 3.11. Крім безпосередньо регістрів в схему включено дешифратор адреси та мікросхеми шинних формувачів (на рисунку умовно не показані).

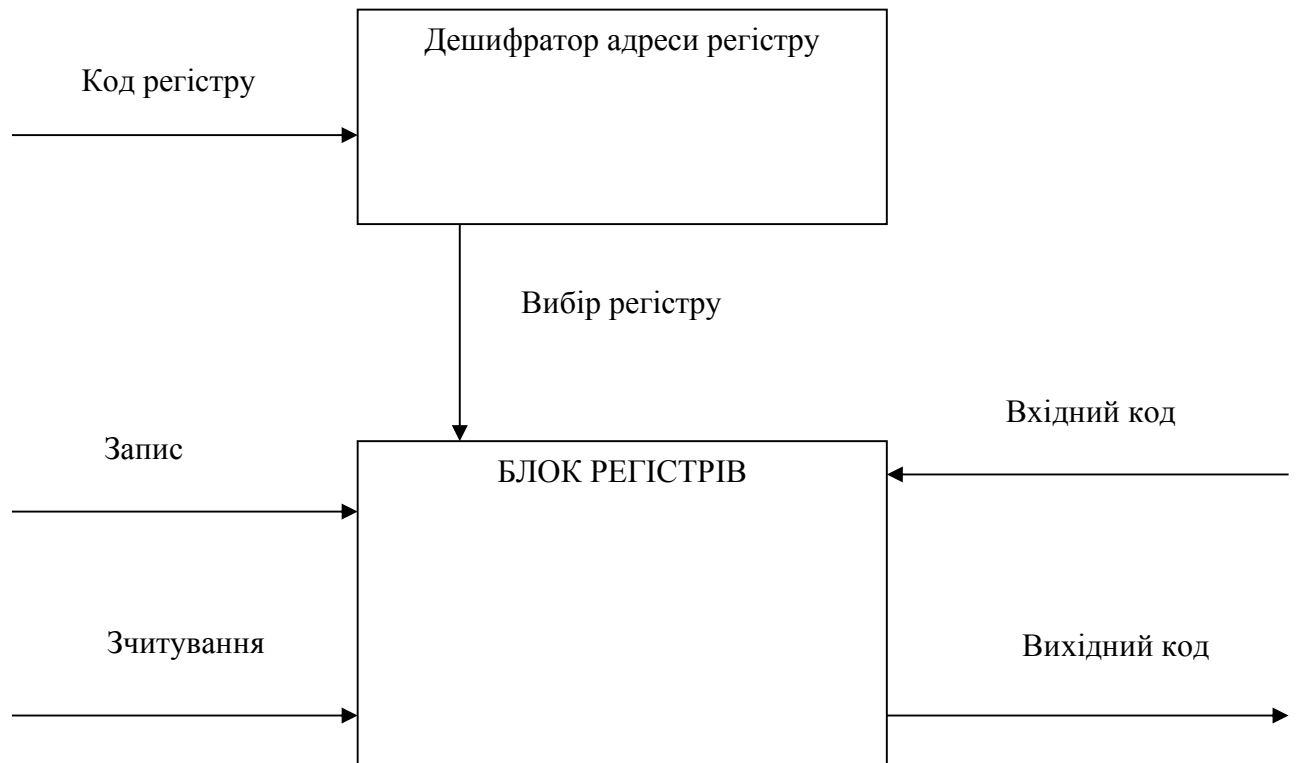


Рисунок 3.11 – Детальна схема регістрового файлу

Схема дешифратора адреси регістру визначається їх кількістю. Тому при її побудові доцільно скористатися пірамідальним методом їх побудови. Під каскадуванням (нарощуванням) розуміють спосіб з'єднання дешифраторів у вигляді мікросхем середнього ступеня інтеграції для одержання більшої розрядності вхідного коду. Приклад з'єднання двох чотирирозрядних дешифраторів типу К555ИДЗ для декодування п'ятирозрядного коду показано на рисунку 3.12.

Мікросхема К155ИДЗ (74154) - дешифратор, що дозволяє перетворити чотирьохрозрядний код, що надійшов на входи F, D, C, D в напругу низького рівня, що з'являється на одному з шістнадцяти виходів 01 ... 016. Дешифратор К155ИДЗ (74154) має два входи дозволу дешифрування G1 і G2. Входи G1 і G2 дозволяють дешифрацію кодів. для цього на них слід подати напругу низького рівня. Ці входи також використовуються при нарощуванні числа розрядів дешифрованого коду. Коли на входах G1 і G2 присутні напруга високого рівня, на виходах 01 ... 015 з'являються високі рівні.

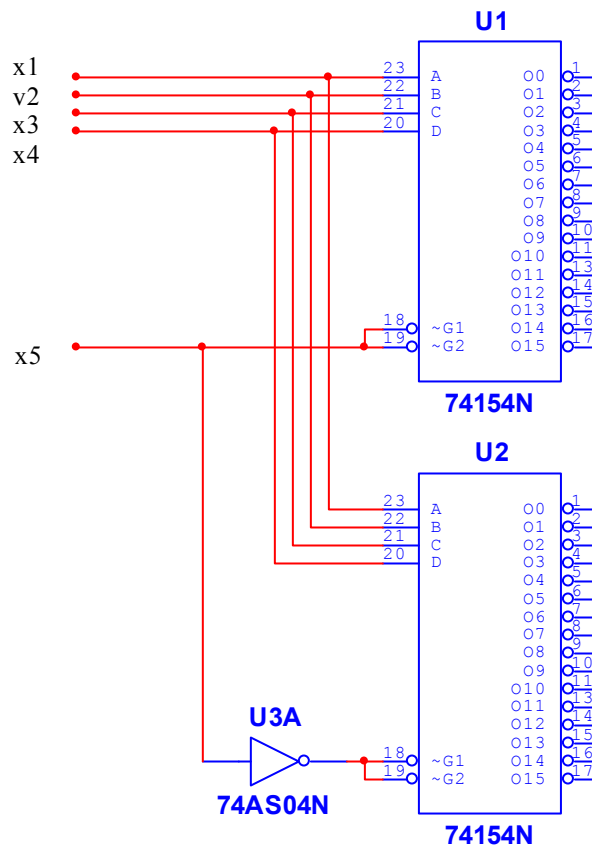


Рисунок 3.13 – Схема пірамідального дешифратора

В обчислювальній техніці досить часто до одного дроту (шини) необхідно підключати багато пристроїв, при дотриманні умови, що в будь-який проміжок часу з цього проводу обмінюватися інформацією будуть тільки два пристрої. Це завдання вирішують за допомогою елементів з трьома станами. На їх основі елементів виготовляються шинні формувачі, умовне позначення якого приведено на рисунку 3.14.

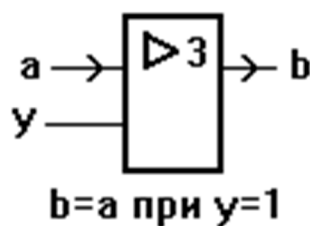


Рисунок 3.14 – Шинний формувач

Інформація передається від «а» до «b» при $y = 1$. Якщо $y = 0$, то елемент знаходиться у високоімпедансному стані.

Синтез складових кластерного регістрового файлу дозволив розробити схему приведену на рисунку 3.15.

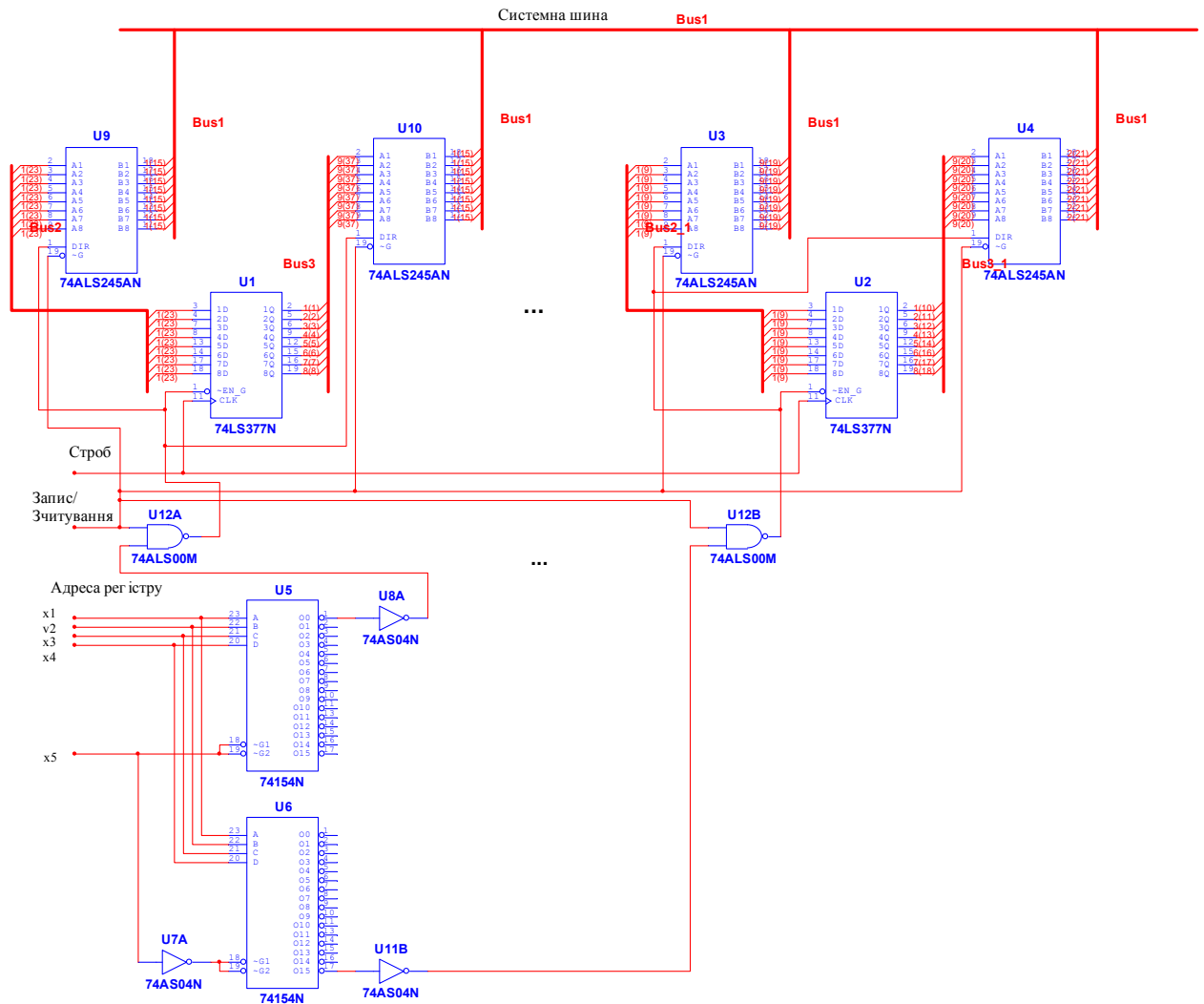


Рисунок 3.15 – Електрична схема кластерного розподіленого регістрового файлу

Таким чином, в даному розділі розроблено та налаштовано в NI Multisim схему кластерного розподіленого регістрового файлу/

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломного проекту є здійснення економічних розрахунків, спрямованих на визначення економічної доцільності апаратного засобу дешифрації за алгоритмом Рабіна. Потрібно визначити доцільність вибраного обладнання, провести розрахунок витрат на розробку даного проектного рішення, визначити прогнозовану ціну апаратного засобу, визначити показники економічної ефективності, зробити відповідні висновки.

4.1 Розрахунок капіталовкладень на розробку апаратного засобу

При загальному підході до розрахунку капіталовкладень, які необхідні на розробку та впровадження апаратного засобу дешифрації, можна записати:

$$K = K_{np} + B_{np} + B_m \quad (4.1)$$

де K – капіталовкладення на створення і впровадження;

K_{np} – витрати на виконання проектних робіт;

B_{np} – кошторисна вартість приладів та обладнання проектованого рішення;

Основними факторами при розрахунку витрат на виконання проектних робіт, що впливають на суму є: затрати часу на виконання проекту, необхідна кількість спеціалістів, їхня заробітна плата.

4.1.1 Розрахунок витрат на оплату праці

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

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

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн.
Керівник ДП, викладач	6026
Консультант техніко-економічного розділу, доцент	6026
Студент	1100

Витрати на оплату праці розробників проекту визначаються за формулою (4.1):

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij} \quad (4.1)$$

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.,

Годинну ставку працівника можна розрахувати за формулою:

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

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

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

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

Коефіцієнт h , який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 1,47.

Середня годинна ставка керівника та консультанта техніко-економічного розділу бакалаврської роботи дорівнює:

$$C_{ij} = \frac{5470 \cdot (1 + 1,47)}{168} = 80,42 \text{ грн./год.}$$

Середня годинна оплата студента дорівнює:

$$C_{ij} = \frac{1200}{168} = 7,14 \text{ грн./год.}$$

Витрати на оплату праці складають:

$$B_{оп} = 20,5 \cdot 80,42 + 2 \cdot 80,42 + 144 \cdot 7,14 = 2837,45 \text{ грн.}$$

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 ДП, доцент	16	80,42	1648,61
Консультант техніко-економічного розділу, доцент	2	80,42	160,84
Студент	144	7,14	1028
Разом			2837,45

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

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 єдиний соціальний внесок складає 16,4% від суми заробітної плати:

$$B_{\phi} = 0,164 \cdot B_{\text{оп}}$$

$$B_{\phi} = \frac{16,4}{100} \cdot 2837,45 = 465,34 \text{ грн.}$$

4.1.3 Розрахунок витрат на матеріали та комплектуючі

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

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

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.

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
М/сх. середньої складності	шт	18	45	810	81	891
М/сх. логіки	шт.	5	10	50	5	55
Папір (формат А4)	уп	2	80	160	16	176
Р а з о м						1041

4.1.4 Витрати на використання комп'ютерної техніки

Витрати на використання комп'ютерної техніки складаються з витрат на амортизацію комп'ютерної техніки, витрат на користування програмним забезпеченням, витрат на електроенергію, що споживається комп'ютером. За даними обчислювального центру ТНЕУ для комп'ютера типу IBM PC/ATX вартість години роботи дорівнює 5,23 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

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	313,8
Оформлення техніко-економічного розділу	8	41,84
Оформлення ДП	12	62,76
Разом	80	418,4

Якщо для розробки КС купується і монтується спеціальне обладнання, то необхідно врахувати також витрати на доставку і монтаж. Ці витрати (в залежності від складності монтажу) можуть бути прийняті у розмірі 10-25% від витрат на придбання обладнання.

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.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1,5 \cdot 2837,45 = 4256,17 \text{ грн.}$$

4.1.6 Інші витрати

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

$$I = 2837,45 \cdot 0,1 = 283,75 \text{ грн.}$$

Витрати на розробку проектного рішення дорівнюють:

$$K_{ПР} = V_{ОП} + V_{\Phi} + V_{М} + V_{ЕЛ} + H + I,$$

$$K_{ПР} = 2837,45 + 465,34 + 510,82 + 418,4 + 4256,17 + 283,75 = 8771,93 \text{ грн.}$$

На підставі отриманих даних за окремими статтями складається кошторис витрат на розробку КС за формою, наведеною в таблиці 4.5.

Таблиця 4.5 - Кошторис витрат на розробку, відлагодження та дослідну експлуатацію КС

Статті витрат	Сума, грн.
1. Матеріальні витрати, в тому числі: матеріали	1041
електроенергія	418,4
2. Витрати на оплату праці	2837,45
3. Відрахування на соціальні потреби	465,34
4. Накладні витрати	283,75
5. Інші витрати.	4256,17
РАЗОМ по кошторису	9030

4.2 Визначення прогнозованої ціни

Величина можливої (договірної) ціни КС повинна визначатися з урахуванням ефективності, якості і термінів її виконання на рівні, що відповідає економічним інтересам замовника (споживача) і виконавця. Договірна ціна (C_d) для прикладних КС розраховується за формулою:

$$C_d = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.4)$$

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

p - середній рівень рентабельності КС, % (приймається в розмірі 20-30% за погодженням з керівником).

$$C_d = 9030 \cdot 1.3 = 11739 \text{ грн.}$$

4.2.1 Економічне обґрунтування вибору комплексу технічних і програмних засобів

Для впровадження більшості КС необхідно:

- придбання та встановлення засобів комп'ютерної техніки;
- придбання та інсталяція системного програмного забезпечення;
- інсталяція і адаптація спеціалізованого програмного забезпечення

Кожен з перерахованих пунктів допускає безліч різних варіантів, так як існує велика кількість конфігурацій комп'ютерів, обладнання та різноманітних програмних продуктів. Кожен з варіантів передбачає різні за величиною і структурою витрати.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

4.3 Розрахунок зведених економічних показників

Економічна ефективність – це співвідношення між отриманим прибутком та затраченими коштами. Вона обчислюється за формулою (4.6):

$$E_{\phi} = \Pi_p / K_B \quad (4.6)$$

де Π_p – очікуваний прибуток ;

K_B – кошторисна вартість.

Очікуваний прибуток можна розрахувати із співвідношення:

$$\Pi_p = C_d - K_B.$$

$$\Pi_p = 11739 - 9030 = 2709 \text{ грн.}$$

Після проведених розрахунків отримуємо:

$$E_{\phi} = 2709 / 9030 = 0.3$$

Термін окупності додаткових капітальних вкладень визначається як :

$$T = 1 / E_{\phi} = 1 / 0.3 = 3.3 \text{ роки.} \quad (4.7)$$

Таблиця 4.6 - Зведені економічні показники розробки

Показник	Значення
Собівартість, грн.	9030
Плановий прибуток, грн.	2709
Ціна, грн.	11739
Економічна ефективність	0,3
Термін окупності, рік	3,3

Провівши аналіз розрахованих значень економічних показників робимо висновок, що розробка апаратного засобу дешифрації за алгоритмом Рабіна є економічно доцільною.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		56

ВИСНОВКИ

В результаті виконання дипломного проекту на тему "Надоперативна пам'ять на основі архітектури кластерного розподіленого реєстрового файлу" отримано наступні результати:

1. сучасні тенденції циркуляції інформаційних потоків показали актуальність застосування симетричних та несиметричних криптосистем;
2. аналіз методів побудови апаратно-програмних систем захисту показав трудомісткість, але разом з тим перспективність апаратної реалізації;
3. здійснено обґрунтування вибору напрямку дослідження та здійснена постановка задачі постановка задачі;
4. розроблено структура апаратного модуля дешифрації та її основних компонентів;
5. здійснена схеми апаратна реалізація та верифікація електричних схем знаходження модуля числа, підкореневого виразу та знаходження кореня;

Техніко-економічне обґрунтування показало економічну доцільність розробки лабораторного стенду дослідження імпульсних блоків живлення.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до курсового проектування з дисципліни “Цифрова схемотехніка”/ Уклад.: І.І.Чикотило, В.Г.Кутуза. – Харків: НТУ“ХПІ”, 2007 .

2. Бабич М.П. Комп’ютерна схемотехніка: Навчальний посібник. / М.П. Бабич, І.А. Жуков – К.: «МК-Прес», 2004.- 412с.

3. Городецкий А. Введение в технологии JTAG и DFT - М.: Palmarium Academic Publishing. 2012. – 308с.

4. Рябенський В.М. Цифрова схемотехніка: Навчальний посібник. / В.М. Рябенський, В.Я. Жуйков, В.Д. Гулий. – Львів: «Новий світ - 2000», 2009. -736с.

5. Угрюмов Е.П. Цифровая схемотехника. / Е.П.Угрюмов – СПб: «БХВ – Петербург», 2004. – 816с.

1. 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, Р.Б.Трембач, Г.М.Мельник, Ю.М.Батько, С.В.Івасьєв / Під ред. О.М.Березького. – Тернопіль: ТНЕУ, 2016. – 65с.

2. Методичні вказівки до написання техніко-економічного розділу для дипломних проектів на здобуття освітньо - кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп’ютерна інженерія» / І.Р.Паздрій. - Тернопіль: ТНЕУ, 2015. – 36с.

3. Бабич М.П., Жуков І.А. Комп’ютерна схемотехніка: Навчальний посібник.- К.:МК-Прес, 2004.-412с

4. Бойко В.І. Схемотехніка електронних систем. Цифрова схемотехніка. Підручник / В.І. Бойко, А.М. Гуржій, В.Я Жуйков - К.:Вища школа, 2004.-423с.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
						58
Зм.	Арк.	№ докум.	Підпис	Дата		

5. Прянишников В.А. Электроника: Полный курс лекций. – СПб.Корона принт; М.: Бином – Пресс, 2006.- 416с.
6. Терехин В.Б., Соловьев Ю.А. Моделирование электронных схем в программе Electronics Workbench. Ч. 1. Создание схем. Ч.2. Элементная база: лабораторный практикум. – Северск: СТИ ТПУ, 2000. – 244 с.
7. Шило В.Л. Популярныe цифровые микросхемы: Справочник. 2 - е изд., испр. – Челябинск: Металлургия, Челябинское отд., 1989.- 352 с.
8. Лещенко М.Є. Основи мікроелектроніки / М.Є. Лещенко, В.Є. Овчаренко. – Х. : Нац. аерокосм. ун-т „Харк. авіац. ін-т”, 2005. – 312с.
9. Комп'ютерна електроніка: Навч. посібник. Частина I/II А.П.Оксанич, С.Е.Притчин, О.В.Вашерук.- Харків: "Компанія СМІТ", 2006.- С200 - 256.
10. Рябенкий В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. Посібник. - Львів: Видавництво «Новий світ 2000», 2009.-736с.
11. Резисторы, конденсаторы, трансформаторы, дроссели, комутационные устройства РЭА: Справочник./ Н.Н. Акимов, Е.П. Ващуков, В.А. Прохоренко, Ю.П. Ходоренок. – Мн.:Беларусь, 1994.- 591с.
12. Токхейм Р. Основы цифровой электроники.- М.:Мир, 1989.
13. Хернитер М. Е. MULTISIM. Современная система компьютерного моделирования и анализа схем электронных устройств. – М.: Издательство: ДМК-Пресс. – 2009. – 409с.
14. Марченко А. Лабораторный практикум по электротехнике и электронике в среде Multisim. Учебное пособие для вузов. / А. Марченко, С. Освальд. – М.: НИЯУ МИФИ, 209. – 364 с.
15. Алексенко А. Г. Основы микросхемотехники — М.: Юнимедиастайл, 2009. — 448 с.
16. Никитин В.А. Схемотехника интегральных схем ТТЛ, ТТЛШ и КМОП: Учебное пособие. М.: НИЯУ МИФИ, 2010. – 64 с.

					ДП.КСМ.07133/14.00.00.000ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59