Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Розрахунково-графічна робота**

з дисципліни

**«**Архітектура комп’ютерів»

по темі:

**«Розробка процесорного ядра»**

Виконав: студент 3 курсу

ФІОТ групи ІО-93

Мамедієв Б.М.

Номер ЗК: 9311

Перевірила: Ткаченко В.В.

Київ 2011

**Технічне завдання**

**на розрахункову роботу**

Розробка процесорного ядра

№ залікової книжки : 9311 10= 100011100111002

|  |  |
| --- | --- |
| Вибір елементної бази | Cyclone II, EP2C35F672C6 |
| Спосіб адресації мікрокоманд | Відносний |
| Структура ПМК | Лінійна |
| Ємність ПМК | 64 слова |
| Призначення  зони β4 | Перевірка на непарність |
| Тривалість  мікрооперації  підсумовування | 4 такта |
| Спосіб множення | Четвертий |
| Розрядність операндів | 8 біт |
| Спосіб мікропрограмування | Горизонтальний; |

Завдання видав Ткаченко В.В.

Завдання прийняв Мамедієв Б.М.

**Зміст**

Вступ…………………………………………………………………………………………..……..4

1. Огляд існуючих рішень……………………………………………………………………...……7

1.1 Принцип структурної організації програмованих логічних інтегральних схем…………...7

1.2 Особливості проектування цифрових пристроїв на базі ПЛІС ………………………...12

1.3 Мікропрограмне управління………………………………………………………………...…14

1.4 Класифікація блоків управління…………………………………………………………….…15

1.5 Опис роботи БМУ………………………………………………………………………………18

2. Розробка БМУ, АЛУ та операційного пристрою………………………………………………20

2.1 Опис АЛП………………………………………………………………………………….……20

2.1.1 Опис математичної складової арифметико-логічного пристрою……………………….…20

2.1.2 Операційна схема…………………………………………………………………………..…21

2.1.3 Змістовний мікроалгоритм………………………………………………………………...…22

2.1.4 Таблиця стану регістрів…………………………………..………………………………..…22

2.2.1 Опис БМУ…………………..…………………………………………………………………23

2.2.2 Структурна схема БМУ………………………………………………………………………24

3. Набір, відлагодження та симуляція роботи розроблюваного пристрою ……………….……25

3.1 Опис роботи схеми АЛП………………………………………………………………….……27

3.2 Опис роботи схеми БМУ…………………………………………………………………….…32

3.3 Схеми АЛП і БМУ………………………………………………………………………………30

4. Тестування розробленого пристрою на апаратному відлагоджувальному комплексі………33

4.1 Розмітка пінів……………………………………………………………………………………34

4.2 Програмування ПЛІС…………………………………………………………………………...35

5. Висновок………………………………………………………………………………………….36

6. Список літератури: ………………………………………………………………………………37

ВСТУП

Програмовані логічні інтегральні схеми - ПЛІС є одними з найбільш перспективних елементів цифрової схемотехніки. ПЛІС являє собою кристал, на якому розташована велика кількість простих логічних елементів. Спочатку ці елементи не сполучені між собою. З'єднання елементів (перетворення розрізнених елементів в електричну схему) здійснюється за допомогою електронних ключів, розташованих в цьому ж кристалі. Електронні ключі управляються спеціальною пам'яттю, в осередки якої заноситься код конфігурації цифрової схеми. Таким чином, записавши в пам'ять ПЛІС певні коди, можна зібрати цифровий пристрій будь-якого ступеня складності (це залежить від кількості елементів на кристалі і параметрів ПЛІС). На відміну від мікропроцесорів, в ПЛІС можна організувати алгоритми цифрової обробки на апаратному (схемному) рівні. При цьому швидкодія цифрової обробки різко зростає.Перевагами технології проектування пристроїв на основі ПЛІС є:

* мінімальний час розробки схеми (потрібно лише занести в пам'ять ПЛІС конфігураційний код);
* на відміну від звичайних елементів цифрової схемотехніки тут відпадає необхідність у розробці і виготовленні складних друкованих плат;
* швидке перетворення однієї конфігурації цифрової схеми в іншу (заміна коду конфігурації схеми в пам'яті);
* для створення пристроїв на основі ПЛІС не потрібно складне технологічне виробництво. ПЛІС конфігурується за допомогою персонального комп'ютера на столі розробника. Тому іноді цю технологію називають «фабрикою на столі».

Одним зі світових лідерів з виробництва ПЛІС є фірма Altera. Для створення цифрових пристроїв на основі своїх виробів Altera розробила спеціальну програмну середу Quartus II. Це середовище дозволяє:

* за допомогою графічного редактора ввести в пам'ять персонального комп'ютера електричну схему;
* перевірити і виправити помилки;
* визначити параметри та характеристики розробленого пристрою;
* сформувати файл конфігурації для конкретної ПЛІС;
* завантажити цей файл в пам'ять інтегральної схеми.

Пояснювальна записка

на розрахункову роботу

Розробка процесорного ядра

# ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

**1.1 Принцип структурної організації програмованих логічних інтегральних схем.**

Будь-яка система ФАЛ (функції алгебри логіки) може бути задана у вигляді ДНФ, тобто у вигляді універсальної структурної записи з конкретизацією в кожному випадку окремих конституент одиниці. Зі сказаного правомірно допустити можливість створення універсальної елементної бази, що реалізує довільну систему ФАЛ цифрового пристрою шляхом завдання певних внутрішніх зв'язків між стандартними базовими логічними елементами І та АБО. Причому доцільним є виконання подібних універсальних елементів у вигляді окремих інтегральних схем.

Для завдання внутрішніх зв'язків між базовими логічними елементами, що дозволяють реалізовувати необхідні ФАЛ, потрібно здійснити попередню настройку таких інтегральних схем. Подібний процес налаштування носить назву програмування. Тому, такі інтегральні схеми отримали назву програмованих логічних інтегральних схем (ПЛІС). Узагальнена структурна схема ПЛІС наведена на рис. 8.1.



рис. 1.1. Узагальнена структурна схема ПЛІС.

Відмінність структури ПЛІС від інших цифрових мікросхем полягає в наявності настроюваних програмним шляхом у загальному випадку безлічі елементів І і АБО, називаються відповідно матрицями І та АБО. Налаштування цих матриць на виконання конкретної ФАЛ здійснюється шляхом подачі на мікросхему ПЛІС спеціальних сигналів програмування, подібних сигналів програмування ПЗУ. У результаті реалізуються внутрішні зв'язки в матрицях, структурно задають з'єднання окремих елементів І і АБО у відповідність з описує роботу цифрового пристрою системі ФАЛ. Вхідні і вихідні змінні подаються на ПЛІС і знімаються з неї через буферні схеми, які, як правило, крім стану логічного нуля або логічної одиниці, можуть брати пасивне Z-стан.

Перші ПЛІС як програмованих містили зазвичай обидві матриці - І та АБО. Такі ПЛІС отримали назву програмованих логічних матриць (ПЛМ), або в англомовній абревіатурі - PLA (Programmable Logic Array). Фрагмент структури ПЛМ, що відображає тільки програмовані матриці І і АБО, представлений на рис. 8.2.

У структурах ПЛІС косою лінією прийнято виділяти наявність електричного з'єднання перетином шин. Якщо коса лінія в перетині відсутній, це означає відсутність електричного зв'язку в перетинаються шинах. Таким чином, зображені на рис. 8.2 програмувальні матриці мають електричні контакти у всіх перетинах. Наявність контактів у всіх перетинах в матрицях характерно для нових, незапрограмованих ПЛІС. Також як і у випадку з ПЗУ, програмування ПЛІС здійснюється руйнуванням електричного контакту в необхідних перетинах, виконаного у вигляді плавкою перемички або транзисторної МДП-структури. Залежно від технічної реалізації цього контакту ПЛІС бувають пропалювальні і перепрограмувальні.

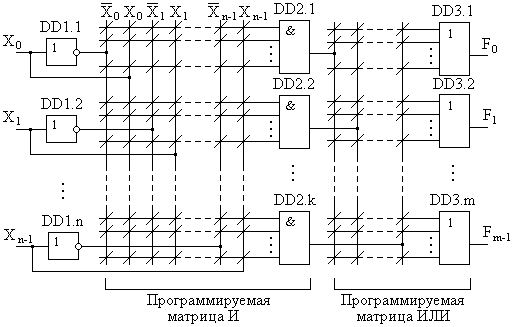


Рис. 1.2. Фрагмент структурной схемы ПЛМ.

Оскільки конституенти одиниці ФАЛ у вигляді ДНФ утворюються як прямими значеннями вхідних змінних, так і інверсними, то і в структурі ПЛІС передбачено наявність вхідних інверторів DD1. Прямі і інверсні значення всіх n вхідних змінних утворюють стовпчики програмованої матриці І. Рядки цієї матриці утворюються n-входові елементами І DD2. У загальному випадку, якщо припустити, що вихідна функція F на всіх наборах вхідних змінних приймає одиничні значення, то таких елементів І за кількістю конституент одиниці повинне бути k = 2n. Синтез будь-якої схеми для такого випадку є недоцільним, оскільки функція з одиничними значеннями на всіх наборах є константою одиниці і від вхідних змінних не залежить. У практичних випадках вихідна функція приймає одиничні значення тільки на певних наборах, а часто і зовсім є частково визначеною. Тому ПЛІС містять зазвичай кількість k елементів І менше, ніж 2n. Виходи цих елементів утворюють k столбців програмованої матриці АБО, а їх вихідні значення відповідають конституентами одиниці функції F. Для можливості реалізації системи ФАЛ з кількох різних вихідних функцій, матриця АБО містить m k-входові елементів АБО DD3. Кожен з цих елементів об'єднує логічним складанням ті конституенти одиниці, які входять до складу ДНФ реалізованої вихідний функції F.

Недоліком структури ПЛМ є слабке використання ресурсів програмованої матриці АБО. Тому була запропонована більш проста, але тим не менш, більш ефективна архітектура програмованої матричної логіки (ПМЛ). В англійській термінології її позначення записується як Programmable Array Logic (PAL). Суть цієї архітектури полягає у використанні як програмованої тільки матриці І. Матриця АБО є фіксованою. Приклад спрощеної структури ПМЛ представлений на рис. 8.3. На малюнку не розкривається структура програмованої матриці І через її аналогією з розглянутим матрицею І в структурі ПЛМ. Вхідні змінні, також як і в ПЛМ, в ПМЛ подаються в прямому і інверсному вигляді. З малюнка видно, що входи елементів АБО DD3 є не комутованими. Таким чином, на кожен вхід поточного елемента АБО подається технологічно певний вихід елемента І програмованої матриці І.

Крім елементів АБО, в зображеної на малюнку структурі ПМЛ містяться буферні елементи DD4, що дозволяють керуючим сигналом на входах E цих елементів, переводити відповідні виходи F ПЛІС в високоімедансний стан. У цьому випадку ці виходи можуть виконувати функції входів програмованої матриці І. З цією метою виходи буферних елементів DD4 з'єднані з матрицею І безпосередньо і через інвертори DD5, тобто організовані зворотні зв'язки. Якщо вихід поточного елемента DD4 знаходиться в активному стані, то через ланцюга зворотного зв'язку значення вихідної функції F подається назад в матрицю І. Це дозволяє реалізовувати схеми не тільки КЦУ, а й ПЦУ. Елементи DD3 і DD4 ПМЛ утворюють так і макрокомірки.

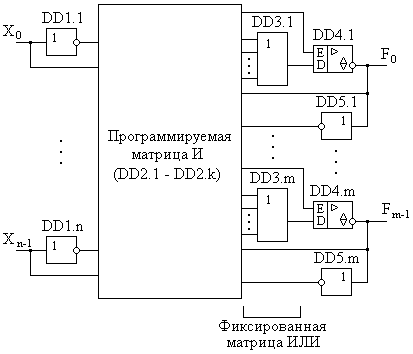


Рис. 1.3. Структурная схема ПМЛ.

Наведена на рис. 8.3. структура характерна для класичних ПМЛ. В даний час більше поширення одержали інтегральні схеми універсальної ПМЛ, принципово відрізняється від класичної тільки структурою макрокомірок. Макрокомірки універсальної ПМЛ містить елемент виключає Або, один вхід якого програмно пов'язаний з нульовим потенціалом загального проводу. Порушення цієї зв'язку дозволяє інвертувати значення вихідних функцій F, змінюючи, таким чином, логічні рівні вихідних сигналів. Крім того, макрокомірки універсальної ПМЛ містять зазвичай D-тригери-засувки, що дозволяють синхронізувати видачу інформації на виходи і мають більш широкі можливості при реалізації ПЦУ.

Іншим представником ПЛІС, інтенсивно розвиваються в даний час, особливо за кордоном, є комплексна ПЛІС. В англійській термінології вона позначається як CPLD (Complex Programmable Logic Devices). Такі ПЛІС містять декілька логічних блоків, кожен з яких представляє універсальну ПМЛ. Входи і виходи всіх логічних блоків об'єднуються в єдину структуру програмованої комутаційної матрицею.

**1.2 Особливості проектування цифрових пристроїв на базі ПЛІС.**

Процес проектування і виготовлення цифрових пристроїв традиційним способом, тобто на основі стандартних інтегральних схем комбінаційного і послідовних типів малої і середньої ступені інтеграції, є вузькоспеціалізованим. Це означає, що фахівці, що беруть участь в процесі створення пристроїв цифрової техніки, виконують певні індивідуальні функції в цьому процесі. Сам процес створення складається з проектних стадій, стадій підготовки виробництва, відпрацювання технології та ін. Аналогічної складністю характеризується і процес створення цифрових пристроїв на основі спеціалізованих інтегральних схем високого ступеня інтеграції. Використання ж ПЛІС дозволяє істотно скоротити обсяг цих стадій, фактично звести їх тільки до етапів проектування з допомогою ЕОМ.

Істотною перевагою ПЛІС є їх універсальність і можливість швидкого програмування під виконання функцій практично будь-якого цифрового пристрою. ПЛІС є напівфабрикат, на основі якого розробник, що володіє персональним комп'ютером, нескладними і відносно недорогими апаратними засобами програмування і спеціальним програмним забезпеченням, званим системою автоматизованого проектування (САПР), має можливість проектування цифрового пристрою в рекордно короткі терміни. При цьому весь цикл проектування та програмування доступний всього одній людині - проектувальнику цифрових пристроїв на базі ПЛІС.

Всі сучасні САПР ПЛІС працюють під управлінням операційних систем сімейства Windows і використовують всі переваги її графічного інтерфейсу. Фірми-виробники інтегральних схем ПЛІС здійснюють звичайно підтримку своєї продукції шляхом розробки та розповсюдження таких САПР. Проектування ПЛІС в системі САПР передбачає виконання таких етапів:

1. розробка структурної формули проектованого цифрового пристрою, її мінімізація та адаптація під тип і можливості використовуваної ПЛІС (для цих цілей використовуються методи, розглянуті в попередніх розділах);
2. створення нового файлу проекту, призначення сімейства ПЛІС для проекту;
3. введення структурної формули або схеми проектованого пристрою за допомогою спеціальних програм - редакторів;
4. компіляція введеної структурної формули або схеми, тобто отримання бітової послідовності завантажується в ПЛІС програми;
5. часовий аналіз затримок в окремих елементах та перевірка працездатності всієї структури цифрового пристрою шляхом моделювання за допомогою спеціальних програм - симуляторів;
6. програмування мікросхеми ПЛІС за допомогою спеціальних апаратних засобів - програматорів.

Проектувальнику в системі САПР доступні зазвичай великі програмні бібліотеки, які визначають функції типових інтегральних схем ТТЛ і КМОП типів. Це дозволяє синтезувати схему цифрового пристрою не тільки в певному базисі, тобто з елементарних функцій І, АБО, НЕ, але і використовуючи готовий програмний аналог існуючих реальних мікросхем. Широкі можливості при проектуванні надає глобальна мережа Internet, через яку здійснюється поширення спеціально розробляються бібліотек компонентів, що не є в стандартних версіях САПР. Для завдання схеми проектованого на базі ПЛІС цифрового пристрою широко використовуються мови опису апаратних засобів, у загальному випадку звані HDL (hardware device language). За допомогою операторів такої мови можна задати типи комбінаційних або послідовних пристроїв, сформулювати вхідні дії на них і зв'язки між ними, визначити константи і змінні в проекті і т.д.

Серед лідерів у виробництві ПЛІС і САПР для них на сьогоднішній день можна виділити такі фірми, як Altera Corporation, Xilinx і Actel Corporation. На їх частку припадає до 80% від загального виробництва ПЛІС, швидкодія яких на сьогоднішній день досягла порядку сотень МГц, ємність програмованих матриць І - кількох мільйонів елементів, а вартість знизилася до десятків у.о. Все це, плюс найкоротші терміни проектування, висока надійність, яка пояснюється високим ступенем інтеграції окремих елементів в єдиному кристалі напівпровідника і, як наслідок, мінімальна кількість зовнішніх міжз’єднань, робить ПЛІС кращою елементної базою в порівнянні зі стандартними логічними мікросхемами комбінаційного і послідовних типів.

## 1.3 Мікропрограмне управління

У більшості сучасних ЕОМ використовується принцип мікропрограмного управління, що отримав розповсюдження в обчислювальній техніці починаючи із 60-х років. Мікропрограмне управління дозволило значно розширити систему команд ЕОМ, так як це досягалося, в основному, за рахунок збільшення об’єму порівняно дешевої пам’яті блоку мікропрограм. У систему команд більшості машин були додані різноманітні варіант інструкцій обробки даних (арифметичних і логічних команд), керування і т.д. Розширення набору команд пояснюється тим, що велика кількість команд машини дозволяв, при його ефективному використанні, скоротити число команд у програмі і цим самим підвищити швидкодію програми. Надалі у багатьох машинах почали розширювати набір команд все складнішими й складнішими командами, намагаючись наблизити машинну мову до рівня мов програмування.

Формат команд у більшості сучасних машин - двухадресний, з різними варіантами адресації (регістр-регістр, регістр-пам’ять,пам’ять-пам’ять)та способами обчислення виконавчої адреси. Це призвело до використання змінної довжини команд (від 2 байтів до 6 і більше).

Вперше усі ці особливості системи команд були реалізовані у архітектурі сімейства IBM360, потім вони перейшли у міні-ЕОМ та мікропроцесори. Але з розвитком мікроелектронних технологій та ускладнення мікропроцесорів, співвідношення між вартістю мікропрограм і обладнання стало змінюватися: вартість блоку пам’яті мікропрограм і блоку схем управління, а також площа, що займається ними на кристалі, мало відрізняються один від одного, а необхідність розміщення пам’яті мікропрограм та іншого обладнання на одному й тому ж кристалі ускладнювали реалізацію мікропрограм великого розміру.

## 1.4 **Класифікація блоків управління**

У мікропроцесорах використовуються два методи видачі сукупності управляючих сигналів: програмний та мікропрограмний.

Виконання операцій в машині зводиться до елементарних претворень інформації (передача інформації між вузлами у блоках, зсув інформації у вузлах, логічні порозрядні операції, перевірка умов тощо) у логічних елементах, вузлах та блоках під дією функціональних керуючих сигналів блоків (пристроїв) керування.

Елементарні перетворення, що не можуть бути розкладені на більш прості, виконуються на протязі одного такту сигналів синхронізації і називаються мікроопераціями.

У апаратних (схемних) пристроях управління кожній операції відповідає свій набір логічних схем, що виробляють певні функціональні сигнали для виконання мікрооперацій у певні моменти часу. При такому способі побудови пристрою управління реалізація мікрооперацій досягається за рахунок жорстко з’єднаних між собою логічних схем, тому ЕОМ з апаратним пристроєм управління називається ЕОМ з жорсткою логікою управління. Це поняття відноситься до фіксації системи команд у структурі зв’язків у ЕОМ і визначає практичну неможливість будь-яких змін у системі команд ЕОМ після її виготовлення.

При мікропрограмній реалізації пристрою управління у склад останнього уводяться ЗП, кожний розряд вихідного коду якого визначає появу певного функціонального сигналу управління. Тому кожній мікрооперації ставиться у відповідність свій інформаційний код - мікрокоманда. Набір мікрокоманд і послідовність їх реалізації забезпечують

виконання будь-якої складної операції. Набір мікрооперацій називають мікропрограмами. Спосіб управління операціями шляхом послідовного зчитування та інтерпретації мікрокоманд з ЗП (найчастіше роль мікропрограиного ЗП виконують швидкодіючі програмовані логічні матриці), а також використання кодів мікрокоманд для генерації функціональних управляючих сигналів називають мікропрограмним, а мікроЕОМ з таким способом управління - мікропрограмними або з логікою управління, що зберігається (гнучкою логікою).

До мікропрограм ставиться ряд вимог щодо функціональної повноти та мінімальності. Перша вимога потрібна для забезпечення можливості розробки мікропрограм будь-яких машинних операцій, а друге пов’язане з бажанням зменшити об’єм потрібного обладнання. Врахування фактора швидкодії приводить до розширення мікропрограм, так як ускладнення останніх дозволяє скоротити час виконання команд програми.

Перетворення інформації виконується в універсальному арифметико-логічному блоці мікропроцесора. Він зазвичай будується на основі комбінаційних логічних схем.

Для прискорення виконання певних операцій вводяться додаткові спеціальні операційні вузли (наприклад, циклічні зсувачі). Крім цього у склад мікропроцесорної системи (МПС) БІС уводяться спеціалізовані оперативні блоки арифметичних розширювачів.

Операційні можливості мікропроцесора можна розширити за рахунок збільшення числа регістрів. Якщо у регістровому буфері закріплення функцій регістрів відсутнє, то їх можна використовувати як для зберігання даних, так і для зберігання адрес.

Подібні регістри мікропроцесора називаються регістрами загального призначення (РЗП). З розвитком технологій реально реалізується виготовлення у мікропроцесорі 16, 32 і більше регістрів.

В цілому ж принцип мікропрограмного управління (ПМУ) включає в себе наступні пункти:

1. будь-яка операція, що реалізується пристроєм, є послідовністю елементарних дій - мікрооперацій;
2. для керування порядком, за яким будуть виконуватися мікрооперації, використовують логічні умови;
3. процес виконання операцій у пристрої описується у формі алгоритму, що представляється у термінах мікрооперацій і логічних умов, і який називається мікропрограмою;
4. мікропрограма використовується як форма представлення функції пристрою, на основі якої визначається структура та порядок функціонування пристрою у часі.

ПМУ надає гнучкості мікропроцесорній системі і дозволяє забезпечити проблемну орієнтацію мікро- та мініЕОМ.

Крім класифікації блоків управління за функціональними ознаками (програмні та мікропрограмні) існують інші:

* централізовані та децентралізовані. У централізованих БУ мікропрограми формуються в одному пристрої для всіх пристроїв у системі. Такі БУ забезпечують виконання усіх мікрооперацій послідовно у часі, що призводить до падіння швидкості системи. У децентралізованих БУ кожен пристрій має свій БУ, роботу яких синхронізує централізований пристрій управління. У різних пристроях можливе виконання мікрооперацій одночасно, що призводить збільшення швидкодії, але збільшує апаратні витрати. У сучасних ЕОМ більш поширені децентралізовані БУ;
* синхронні та асинхронні. У синхронних БУ для виконання кожної мікрооперації виділяються однакові проміжки часу, що дорівнюють максимальній тривалості МО. У асинхронних БУ на кожну МО виділяють стільки часу, скільки вона потребує для виконання. Асинхронні БУ є більш швидкодіючими порівняно із синхронними, але потребують збільшення апаратної складності. На практиці застосовують комбіновані БУ, у яких МО групуються за часом виконання.

## 1.5 Опис роботи БМУ

БМУ функціонує у відповідності з принципом мікропрограмного управління, що полягає у наступному.

Під час виконання мікропрограми в кожному такті із постійної пам’яті БМУ зчитується та розшифровується чергова мікрокоманда. В результаті виконання мікрокоманди формуються управляючі сигнали необхідної тривалості, що поступають на всі функціональні частини обчислювальної системи, а також формується адреса наступної мікрокоманди.

Можна виділити наступні етапи виконання команди в обчислювальній системі:

1. Вибірка команди. З ОП зчитується команда в регістр команд процесора, для чого виконується відповідна МП, що записана у пам’ять БМУ.
2. Розпакування команди. Команда розшифровується (аналізуються поля слова команди, визначаються операнди), що забезпечується виконанням відповідної МП.
3. Виконання операції. Виконується МП виконання заданої операції над визначеними операндами.
4. Формування адреси наступної команди. Відповідна МП формує адресу наступної команди у лічильнику команд.

Спрощена структурна схема БМУ наведена на рис. 1.1.

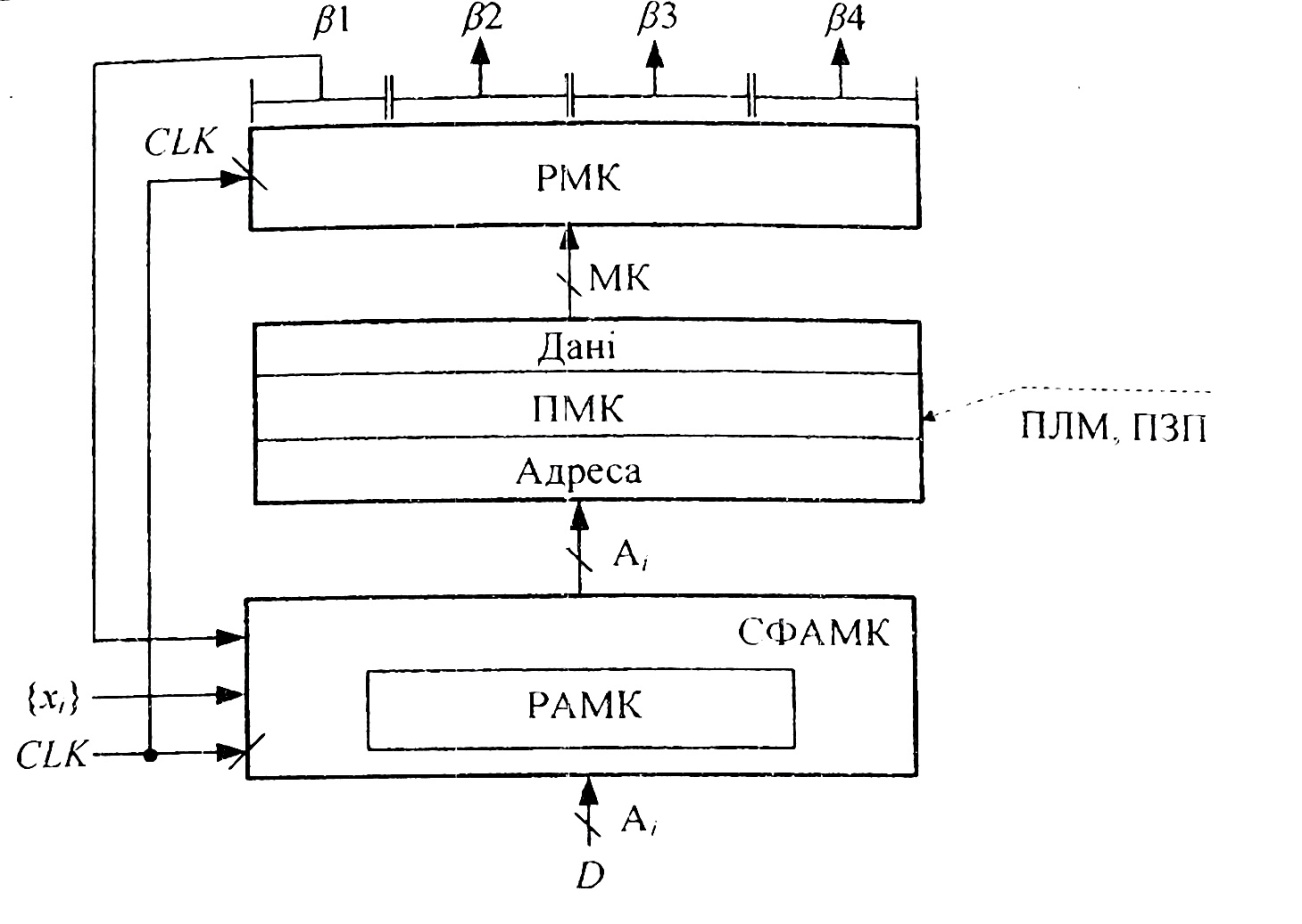


Рисунок 1.4 - Структурна схема БМУ

Основні функціональні частини БМУ:

* РАМК - регістр адреси МК;
* СФАМК - схема формування адреси МК;
* ПМК - пам'ять МК;
* РМК - регістр МК;
* Аі- адреса МК;
* CLK - синхросигнал;
* {xi} - логічні умови;
* D - вхід завдання початкової адреси мікропрограми.

МК розміщується у пам’яті мікропрограм. На рис.1.2 наведений формат мікрокоманди.

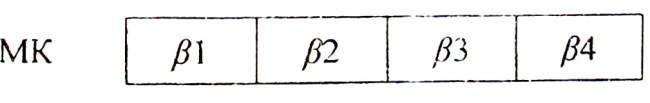


Рисунок 1.5 - Формат мікрокоманди

B1 – зона формування наступної адреси МК;

B2 – зона управляючих сигналів;

B3 – зона визначення тривалості управляючих сигналів;

B4 – зона службових розрядів;

У кожному такті за синхросигналом CLK адреса мікрокоманди поновлюється у РАМК і надходить на адресний вхід ПМК. За адресою, що надійшла у ПМК, обирається відповідна мікрокоманда і видається на вихід даних ПМК. Слово мікрокоманди записується у РМК за зворотнім перепадом синхросигналу CLK.

Сигнали зони β2 управляють вузлами МПС, зони β3 - визначають тривалість цих сигналів, сигнали зони β1 разом із логічними умовами {xi} поступають на вхід СФАМК і формують адресу наступної МК. За черговим сигналом CLK адреса наступної МК буде сформована у РАМК. Зона β4 використовується для виконання допоміжних функцій, наприклад, контролю апаратури.

**2. Розробка БМУ, АЛУ та операційного пристрою**

**2.1 Опис АЛП**

**2.1.1 Опис математичної складової арифметико-логічного пристрою**

За технічним завданням маємо четвертий спосіб множення, числа подані в прямому коді з розрядністю 8 біт.

Під час множення чисел у прямих кодах знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюється підсумовування по модулю 2 цифр, що розміщується в знакових розрядах співмножників.

Множник і множене правильні двійкові дроби вигляду Х=0,х1,х2,…хn, Y=y1,y2,…yn, де x та y належать множині {0,1}. Тоді добуток Z абсолютних величин чисел Х та Y дорівнює

Z=YX=Yx12-1+Yx22-2+…+Yxi2-i+…+Yxn2-n.

Множення двох чисел X та Y може бути реалізоване шляхом використання визначеного циклічного процесу, характер якого залежить від конкретної форми виразу Z. Один цикл множення складається з додавання чергового часткового добутку, що являє собою добуток множеного на одну цифру множника, до суми часткових добутків.

У четвертому способі множення здійснюється зі старших розрядів множника, що зсувається вліво, сума часткових добутків нерухома, а множене зсувається вправо.

**2.1.2 Операційна схема**

**RG1**

**1**

**1**

**1**

**2n**

**SM**

**2n**

**2n**

**RG2(1)**

**RG2**

**n**

**1**

**2n**

**RG3**

**1**

Рисунок 2.1−Операційна схема пристрою множення четвертим способом

**2.1.3 Змістовний мікроалгоритм**

Початок

RG1:=0

RG2:=X

RG3:=Y.0

RG3:=0.r(RG3)

0

RG2[1]

1

RG1:=RG1+RG3

RG3:=0.r(RG3)

RG2:=l(RG2).0

1

RG2=0

0

Кінець

Рисунок 2.2−Змістовний мікроалгоритм пристрою множення четвертим способом

## 2.1.4 Таблиця стану регістрів

|  |  |  |  |
| --- | --- | --- | --- |
| № ц. | RG1 | RG3 | RG2 |
| П.С. | 0000000000000000 | 0111110010000000 | **1**1111001 |
| 1 | 0111110010000000 | 0011111001000000 | 11110010 |
| 2 | + 0011111001000000  = 1011101011000000 | 0001111100100000 | 11100100 |
| 3 | + 0001111100100000  = 1101100111100000 | 0000111110010000 | 11001000 |
| 4 | + 0000111110010000  = 1110100101110000 | 0000011111001000 | 10010000 |
| 5 | + 0000011111001000  = 1111000100111000 | 0000001111100100 | 00100000 |
| 6 | 1111000100111000 | 0000000111110010 | 01000000 |
| 7 | 1111000100111000 | 0000000011111001 | 10000000 |
| 8 | + 0000000011111001  = 1111001000110001 | 0000000001111100 | 00000000 |

## 2.2 Опис роботи БМУ

БМУ функціонує у відповідності з принципом мікропрограмного управління, що полягає у наступному.

Під час виконання мікропрограми в кожному такті із постійної пам’яті БМУ зчитується та розшифровується чергова мікрокоманда. В результаті виконання мікрокоманди формуються управляючі сигнали необхідної тривалості, що поступають на всі функціональні частини обчислювальної системи, а також формується адреса наступної мікрокоманди.

Можна виділити наступні етапи виконання команди в обчислювальній системі:

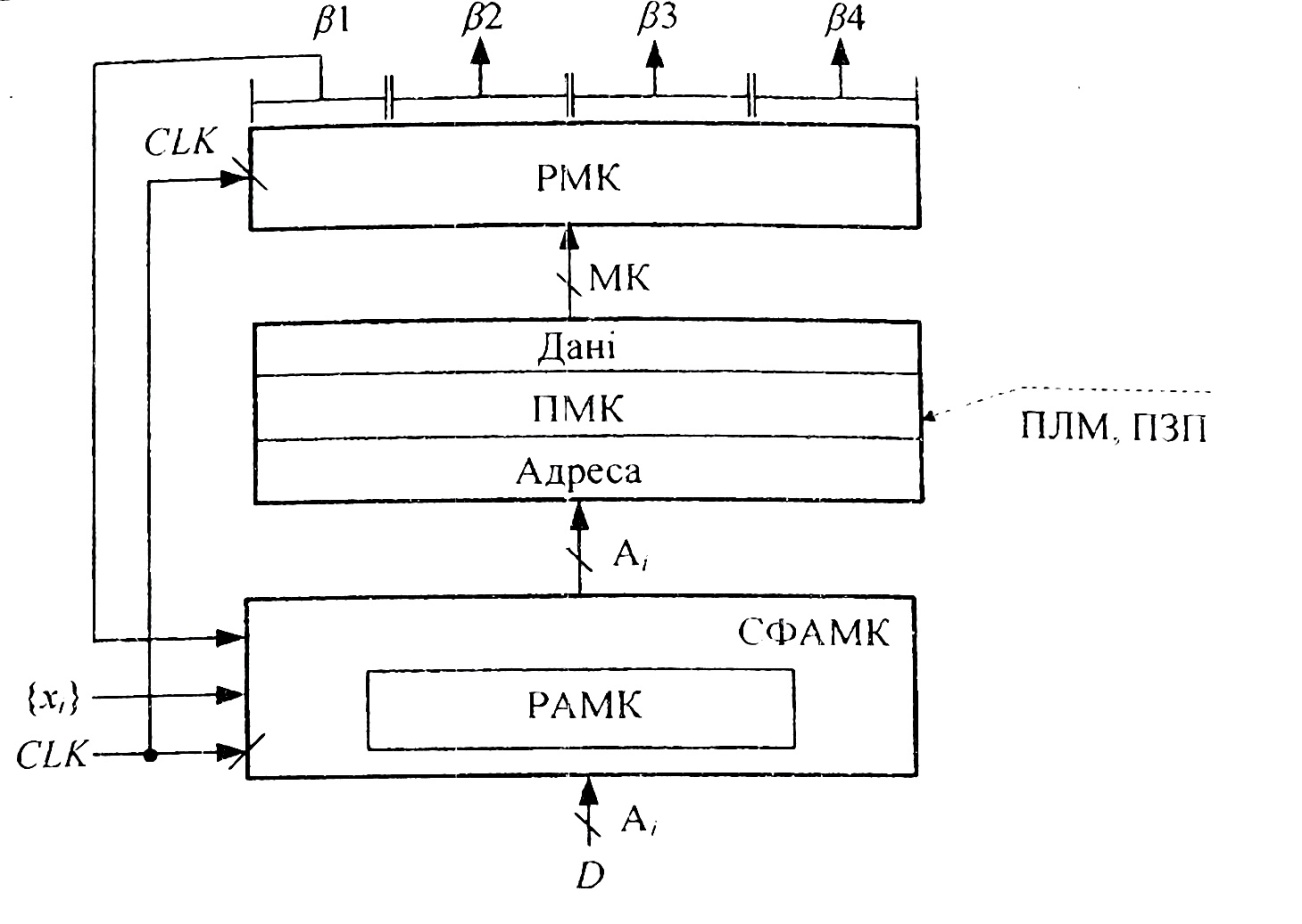
1. Вибірка команди. З ОП зчитується команда в регістр команд процесора, для чого виконується відповідна МП, що записана у пам’ять БМУ.
2. Розпакування команди. Команда розшифровується (аналізуються поля слова команди, визначаються операнди), що забезпечується виконанням відповідної МП.
3. Виконання операції. Виконується МП виконання заданої операції над визначеними операндами.
4. Формування адреси наступної команди. Відповідна МП формує адресу наступної команди у лічильнику команд. Спрощена структурна схема БМУ наведена на рис. 1.4. 

Рисунок 1.4 - Структурна схема БМУ

Основні функціональні частини БМУ:

* РАМК - регістр адреси МК;
* СФАМК - схема формування адреси МК;
* ПМК - пам'ять МК;
* РМК - регістр МК;
* Аі- адреса МК;
* CLK - синхросигнал;
* {xi} - логічні умови;
* D - вхід завдання початкової адреси мікропрограми.

МК розміщується у пам’яті мікропрограм. На рис.1.2 наведений формат мікрокоманди.

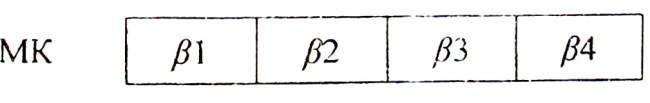


Рисунок 1.5 - Формат мікрокоманди

B1 – зона формування наступної адреси МК;

B2 – зона управляючих сигналів;

B3 – зона визначення тривалості управляючих сигналів;

B4 – зона службових розрядів;

У кожному такті за синхросигналом CLK адреса мікрокоманди поновлюється у РАМК і надходить на адресний вхід ПМК. За адресою, що надійшла у ПМК, обирається відповідна мікрокоманда і видається на вихід даних ПМК. Слово мікрокоманди записується у РМК за зворотнім перепадом синхросигналу CLK.

Сигнали зони β2 управляють вузлами МПС, зони β3 - визначають тривалість цих сигналів, сигнали зони β1 разом із логічними умовами {xi} поступають на вхід СФАМК і формують адресу наступної МК. За черговим сигналом CLK адреса наступної МК буде сформована у РАМК. Зона β4 використовується для виконання допоміжних функцій, наприклад, контролю апаратури.

## 2.2.1 Закодований мікро алгоритм

Закодовані мікрооперації та сигнали приведені у таблиці 2.1.

Таблиця 2.1−Таблиця кодування мікрооперацій

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця кодування мікрооперацій | |  | Таблиця кодування логічних умов | |
| МО | УС |  | ЛУ | Позначення |
| RGZ:=0  RGX:=X.0  RG3Y:=Y  RGZ:=RGZ+RGX  RGX:=0.R.(RG1)  RGY:=L.(RG2).0 | R (y0)  W2 (y0)  W3 (y0)  W1 (y2)  SL1 (y1)  SL2 (y3) |  | RGY(n)  RGY=0 | X1  X2 |
|  |

Закодований функціонально-структурниймікроалгоритм приведений на рисунку 2.4.

## 2.2.2 Розрахунок параметрів БМУ



Рисунок 2.4− Закодований функціонально-структурний мікроалгоритм

Визначимо розрядність адреси ПМК:

Розмістимо мікрокоманди в пам’яті, враховуючи початкову адреса згідно з технічним завданням (8h = 0010002) (рисунок 2.5).

Рисунок 2.5−Розміщення команд у ПМК

**3.2.1 Карта пам’яті мікрокоманд**

|  |  |  |
| --- | --- | --- |
| Адресса команди | Двійкова адресса | Номер команди |
| 8 | 001000 | П (1) |
| 9 | 001001 | - |
| 10 | 001010 | К (5) |
| 11 | 001011 | 2 |
| 12 | 001100 | 3 |
| 13 | 001101 | 4 |

При мінімальному кодуванні управляючих сигналів та горизонтальному програмуванні довжина зони β2 дорівнює кількості управляючих сигналів:

За технічним завданням, тривалість мікрооперації підсумовування становить 4 тактів. Тривалість усіх інших мікрооперацій вважатимемо рівною 1.

Отже, довжина зони β3 становить:

Для отримання необхідної затримки необхідно використати додатковий код часу затримки. Так як усі операції виконуються 1 такт, то :

tз = 3 – 1 =210 =0.102ПК= 1.102ДК.

Для перевірки на парність у зоні β4 необхідно виділити один розряд:

Враховуючи попередні обчислення отримаємо довжину команди:

Складемо таблицю кодування розрядів поля управління мультиплексором (таблиця 2.3).

Карта програмування БМУ зображена у таблиці 2.3, структурна схема БМУ зображена на рисунку 2.6.

**3.2.2 Структурна схема БМУ**

Таблиця 2.3−Карта програмування БМУ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Адрес | В1 | | | | | | Управляючі сигнали | | | | Затримка | | | | | B4 |
| № |  | SM | | | | MS | | y0 | y1 | y2 | y3 | ЗН |  | | |  | |
| П(1) | 1000 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 2 | 1011 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 3 | 1100 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | |
| 4 | 1101 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | |
| 05 | 1010 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |



Рисунок 2.6−Структурна схема БМУ

1. **Набір, відлагодження та симуляція роботи розроблюваного пристрою**

3.1 **Опис роботи схеми АЛП**

АЛП і БМУ данної схеми керуються синхросигналами. Сигнали БМУ керують записом початкових значень в регістри, лічильником, зсувами та додаванням. Для занесення початкових значень в регістр використали елемент константи та мультиплексор, котрий по сигналу початкового стану записує початковий стан регістру, а при його відсутності подається значення з суматора. АЛП формує сигнали х1 та х2, котрі є управляючими сигналами для БМУ

Х1 – старший біт результату

Х2 – логічне «І» всіх розрядів лічильника. При Х2=1 множення закінчується.

Схема АЛУ наведена нижче

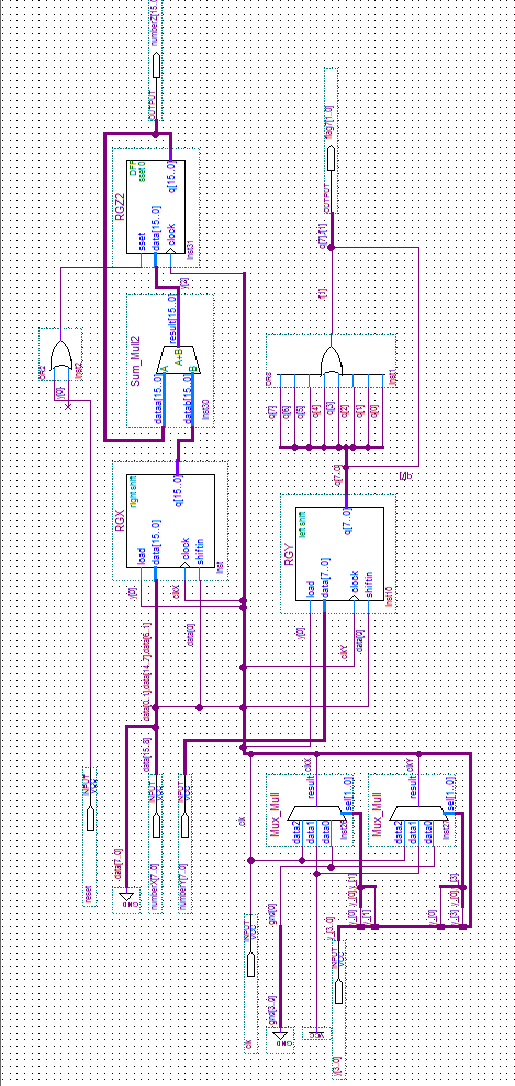
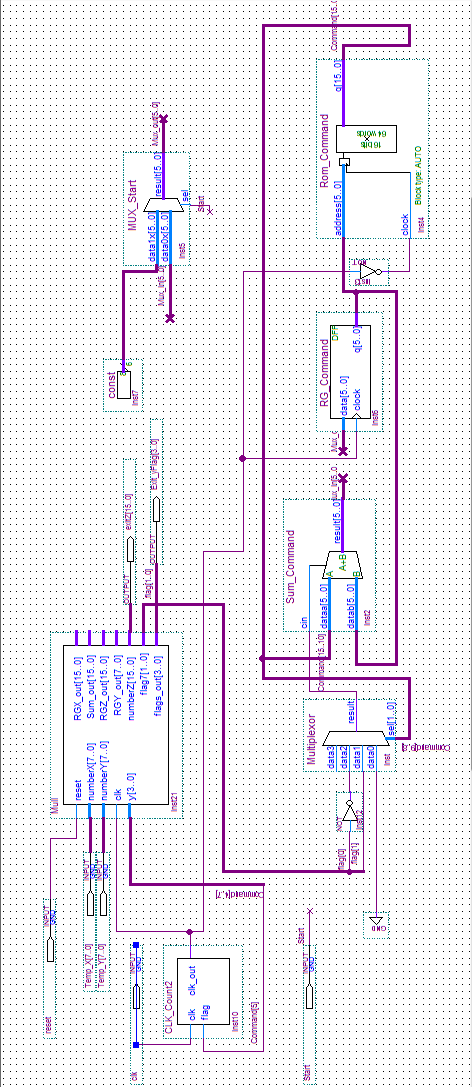
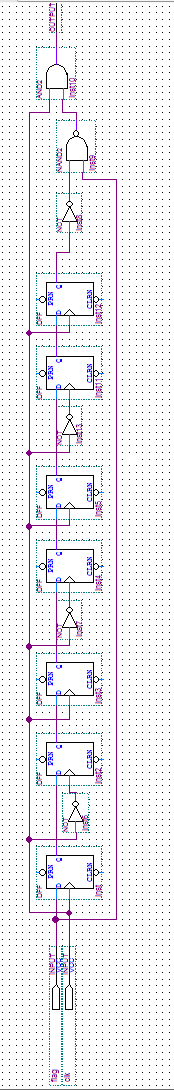


Схема БМУ



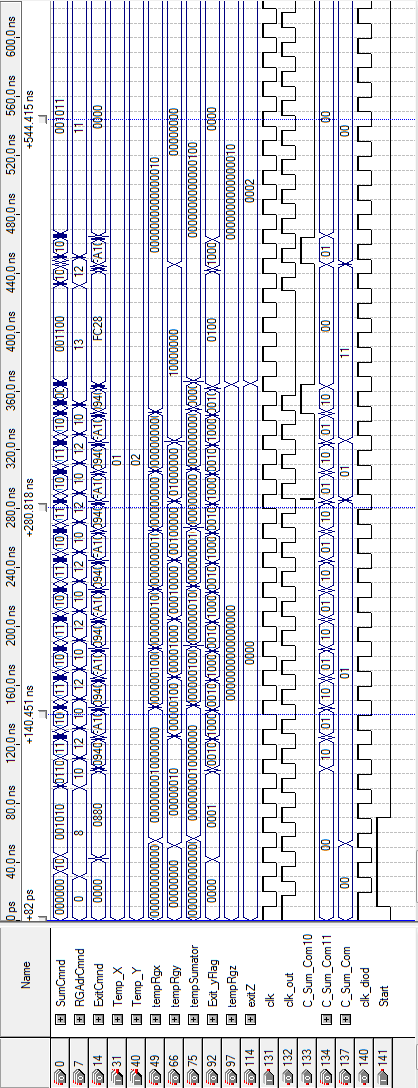


Рисунок 3.3 – Часова диаграмма роботи БМУ

Для конфігурації пам’яті ROM було використано MemoryInitializationFile для вказання вмісту.

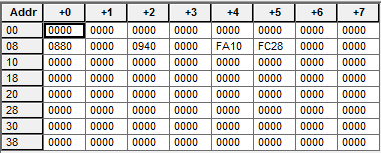


Рисунок 3.4 – Файл памяти

**4.** **Тестування розробленого пристрою на апаратному відлагоджувальному комплексі**

**4.1 Розмітка пінів**

Для того, щоб співставити вхідний/вихідний пін на схемі з піном на ПЛІС, використовують програму PinPlaner.(рисунок 4.1).

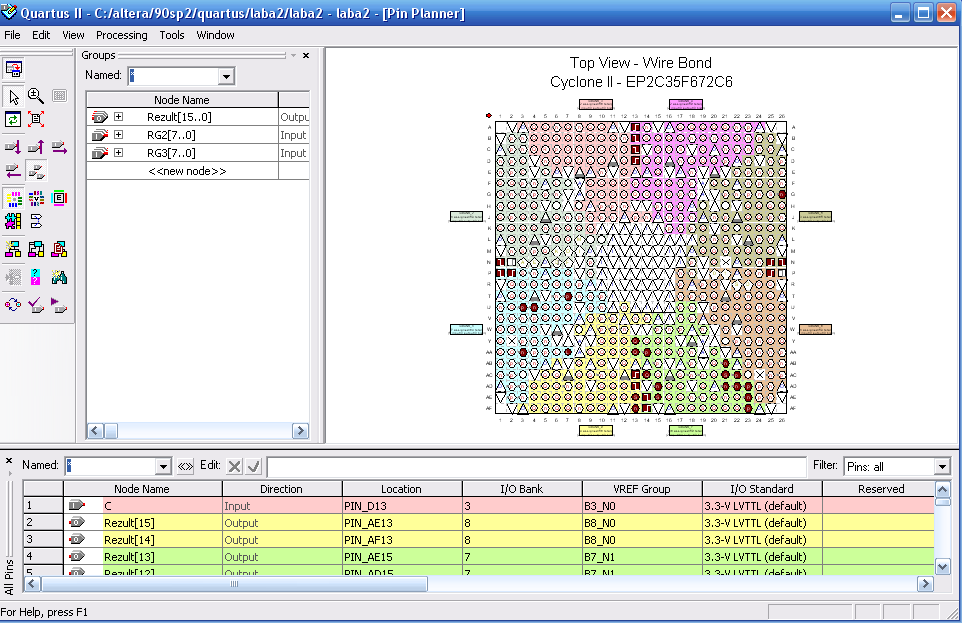


Рисунок 4.1 – Окно программы PinPlaner

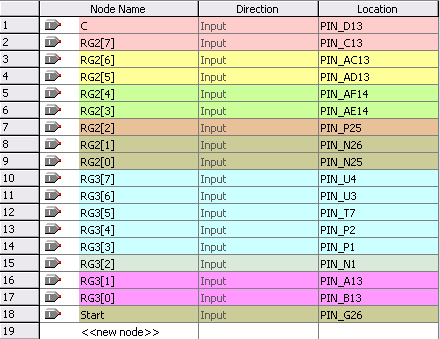


Рисунок 4.2 – Таблица розпиновки входных сигналов

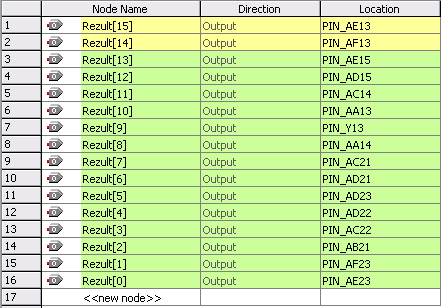


Рисунок 4.3 – Таблица розпиновки выходных сигналов

Результат розпіновки на рис. 4.4

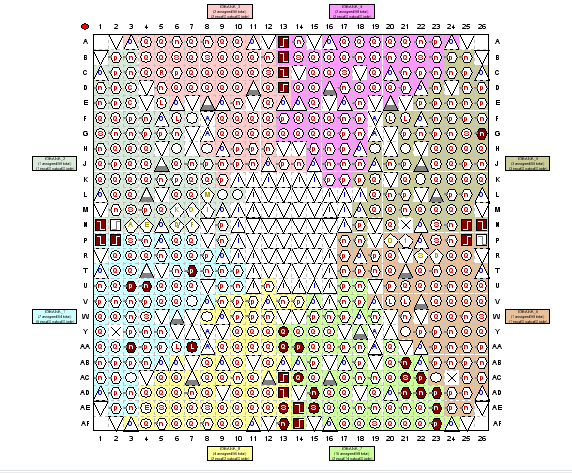


Рисунок 4.4 – Результат розпиновки

**4.2 Програмування ПЛІС**

Програмування ПЛІС проводиться за допомогою утиліти Programmer. В HardWareSetup вибираємо Bluster Devise, натискаючи на кнопку Start починаємо роботу з ПЛІС.

**5.** **Висновок**

В результаті виконання завдання на елементній базі FPGА був створений фрагмент процесорного ядра, що складається з арифметико-логічного пристрою для четвертого способу множення і блоку управління. Були проведені необхідні розрахунки параметрів БМУ та операційного пристрою, приведені відповідні структурні схеми.

Був виконаний набір електричної схеми пристрою у САПР Quartus ||. Отримані схеми перевірено на відсутність помилок та перевірено на правильність роботи. за допомогою побудови часових діаграм роботи пристрою засобами САПР Quartus 2.

Отримані електричні схеми були завантажені у апаратний відлагоджувальний комплекс Altera DE2. Отриманий пристрій працював у відповідності з поставленим завданням.

Процес виконання роботи був докладно описаний.

**6. Список літератури:**

1) Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. – Арифметичні та управляючі пристрої цифрових ЕОМ: Навчальний посібник. – К.:ВЕК+, 2008. – 176 с.

2) Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. – Прикладна теорія цифрових автоматів: Навч. посібник. – К.: Книжкове вид-во НАУ, 2007. – 364 с.

3) User Manual: Altera DE2 Board. - Altera Corporation, 2006. – 72 c.

4) http://uk.wikipedia.org/wiki/Микроконтроллер

5) http://uk.wikipedia.org/wiki/ПЛІС

6) Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. – Мікропроцесорні системи

7) Зиссос Д. Проектирование систем на микропроцессорах/ Пер. с англ. под ред. А.И.Петренко. – К.: Техніка, 1982. – 176 с.1. Савельев А.Я. Арифметические и логические основи цифрових автоматов: Учебник. – М.: Высшая школа, 1980.

8) Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. Л.: Машиностроение, 1977

9) Поспелов Д.А. Логические методы анализа и синтеза схем. – М.: Энергия, 1974.

10) Проектирование цифровых вычислительных машин/ Под ред. С.А. Майорова. Учебное пособие для студентов вузов. – М.: Высшая школа, 1972.

11) Савельев А.Я. Арифметические и логические основы цифровых автоматов. – М.: Высшая школа, 1980.

12) Самофалов К.Г., Корнейчук В.И., Тарасенко В.П., Жабин В.И. Цифровые ЭВМ. Практикум. – К.: Вища школа., 1990.

13) Скляров В.А. Синтез автоматов на матричных БИС/ Под ред. С.И.Баранова. – Минск: Наука и техника, 1984.