Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**Розрахунково-графічна робота**

з дисципліни

**«Архітектура комп’юторів»**

по темі:

**«Розробка процесорного ядра»**

Виконав: студент 3 курсу

ФІОТ групи ІО-91

Скорик Д.В.

Номер ЗК: 9112

Перевірив: Ткаченко В.В.

**Київ 2011**

**Технічне завдання**

**на розрахункову роботу**

Розробка процесорного ядра

Студ**.** Скорик Д.В. гр. ІО-91

№ залікової книжки : 9112 10= 100011100110002

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

Завдання видав Ткаченко В.В.

Завдання прийняв Скорик Д.В

**Зміст**

[Вступ…………………………………………………………………………….…](#_Toc295298779)4

[1 Огляд існуючих рішень……………………………………………..………….](#_Toc295298780) 6

[1.1 Мікропрограмне управління…………………………………………………](#_Toc295298781) 6

[1.2 Класифікація блоків управління……………………………………………..](#_Toc295298782) 6

[1.3 Опис роботи БМУ…………………………………………………………….](#_Toc295298783) 8

[2. Розробка БМУ,АЛУ та операційного пристрою……………….……………4](#_Toc295298784)1

2.1 Опис АЛП……………………………………………………………………..11

[2.1.1 Опис математичної складової арифметико-логічного пристрою.............4](#_Toc295298785)1

[2.1.2 Операційна схема…………………………………………………………..4](#_Toc295298786)1

[2.1.3 Змістовний мікроалгоритм………………………………………………… 4](#_Toc295298787)2

[2.1.4 Функціональна схема арифметико-логічного пристрою………………… 4](#_Toc295298788)3

[2.1.5 Закодований мікроалгоритм……………………………………………….. 4](#_Toc295298789)4

2.1.6 Опис роботи схеми АЛП…………………………………..………………..15

2.2 Опис БМУ…………………………………………………..………………….17

[2.2.1 Розрахунок параметрів БМУ………………………………………………. 4](#_Toc295298790)7

2.2.2 Опис роботи схеми БМУ…………………………..………………………..19

2.3 Опис операційного пристрою…………………………..…………………….21

2.3.1 Опис роботи операційного пристрою…………………….………………..21

[3 Набір, відлагодження та симуляція роботи розроблюваного пристрою…………………………………………………………………………...4](#_Toc295298791)4

[3.1 Загальні відомості…………………………………………………………….](#_Toc295298792) 24

[4. Тестування розробленого пристрою на апаратному відлагоджувальному комплексі………………………………………………..…………………………2](#_Toc295298798)5

[4.1 Розмітка пінів ………………………………………………………………….2](#_Toc295298799)5

[4.2 Програмування ПЛІС…………………………………………………………](#_Toc295298801) 26

5.[Висновки…………………………………………………………………...…….](#_Toc295298802)27

6.[Список літератури …………………………………………………..……](#_Toc295298804)……..28

**ВСТУП**

У наш час значно зросла значимість побудови обчислювальних машин на сучасній елементній базі,дозволяють проектувати перспективні архітектури машин майбутніх поколінь. Технологія ПЛІС дозволяє за короткий проміжок часу, без використання фінішних технологій, реалізувати в кристалі практично будь-який проект цифрового пристрою, маючи в наявності лише персональний комп'ютер і САПР ПЛІС. Тому створення перспективних засобів обчислювальної техніки, не прив'язаних до технологічних ліній виробництва, що в цей же час є гнучкими, на основі сучасних кристалів ПЛІС представляє актуальну науково-технічну проблему.

Гнучкі системи на базі ПЛІС широко використовуються в багатьох областях: гнучка обробка даних; цифрова обробка сигналів; обробка зображень; комунікації; обчислювальні пристрої загального призначення; верифікація.

ПЛІС - програмована логічна інтегральна схема, що поєднує регулярність структури напівпровідникового запам’ятовуючого пристрою з універсальністю мікропроцесора, що дозволяє програмно формувати внутрішній спеціалізований процесор. Структурно ПЛІС є однорідною середою і має властивості однорідності, гнучкості і паралельності виконання операцій. Паралельність а отже і підвищення швидкодії, досягається за рахунок підвищення тактової частоти і за рахунок паралельного виконання великої кількості операцій. Надійність, гнучкість і структурна універсальність (можливість створення для кожного завдання відповідної структури) забезпечуються апаратно за рахунок програмної зміни зв'язків між елементами і функцій самих елементів. Однорідність - простота технології виготовлення при використанні однакових елементів і однотипних зв'язків між ними.

Пояснювальна записка

**на розрахункову роботу**

Розробка процесорного ядра

# 

# 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

## 1.1 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.

## 1.2 **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 мікропрограм ній 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.

## 1.3 Опис роботи БМУ

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.

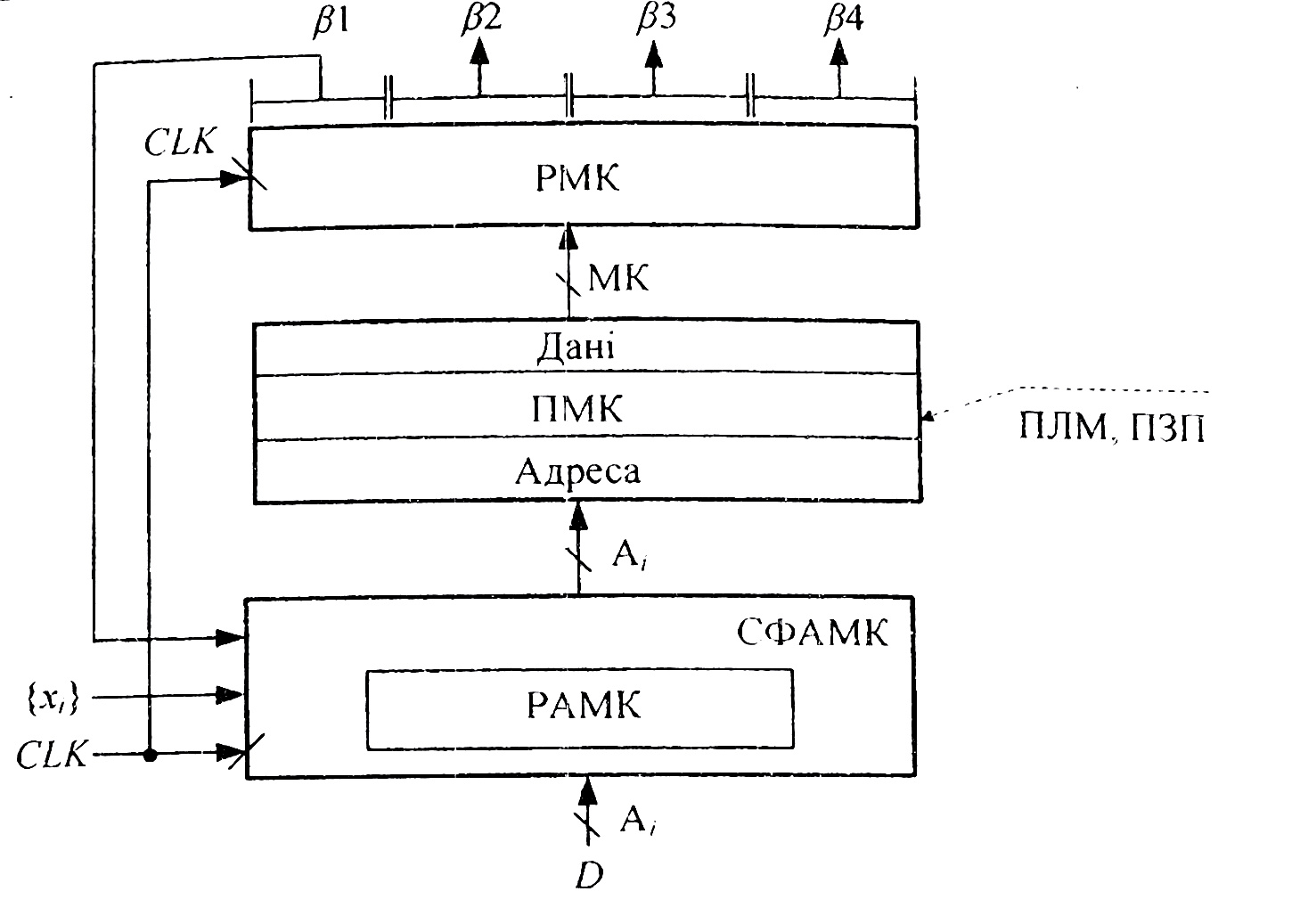


Рисунок 1.1 - Структурна схема БМУ

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.

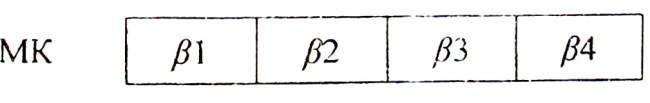


Рисунок 1.2 - Формат мікрокоманди

Основним призначенням СФАМК є реалізація управляючих структур, що зустрічаються у мікропрограмах: лінійна послідовність, структури виду «якщо А, то Б, інакше В» та структури виду «поки А, роби Б». При цьому схема виконує наступні функції:

* проводить дешифрацію коду операції команди (КОП) для звернення до першої мікрокоманди мікропрограми, що інтерпретує дону команду;
* формує адреси наступних мікрокоманд;
* зберігає ознаки переходів, до виходять з операційного блоку то створюються при виконанні мікрокоманд умовного переходу;
* виконує управління перериваннями на мікропрограмному рівні.

Пам'ять мікропрограм призначена для зберігання мікрокоманд, її ємність та розрядність однозначно визначаються набором мікропрограм, що реалізуються. Шляхом зміни набору мікропрограм можна змінювати систему команд мікропроцесора і тим самим орієнтувати його функціональну спрямованість.

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 синхросигналуCLK.

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**

Виходячи з ТЗ, множення виконується другим способом.Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою, множене зсувається праворуч, множник ліворуч.

Вираз

подамо у вигляді



.



Отже, сума часткових добутків в i-му циклі, де , зводиться до обчислення виразу:

.



з початковими значеннями i=1, Y0=2-nY, Z0=0.

**2.1.2 Операційна схема**

Операційна схема арифметико-логічного пристрою для обчислення функції зображена на рисунку 2.1. Призначення регістрів: RG1 – результат обчислення Z = X\*Y, RG3 - значення Y, RG2 – значення X.



Рисунок 2.1−Операційна схема пристрою множення другим способом

**2.1.3 Змістовний мікроалгоритм**

Змістовний мікроалгоритм зображено на рисунку 2.2.



Рисунок 2.2−Змістовний мікроалгоритм пристрою множення другим способом

**2.1.4 Функціональна схема арифметико-логічного пристрою**

Функціональна схема зображена на рисунку 2.3.



Рисунок 2.3−Функціональна схема арифметико-логічного пристрою для обчислення функції

**2.1.5 Закодований мікроалгоритм**

Закодовані мікрооперації та сигнали приведені, відповідно у таблицях 2.1 та 2.2.

Таблиця 2.1−Таблиця кодування мікрооперацій

|  |  |
| --- | --- |
| Мікрооперації | Управляючі сигнали |
| CLR | y1 |
| WR2, WR3 | y2 |
| WR1 | y3 |
| SL, SR | y4 |

Таблиця 2.2−Таблиця кодування сигналів

|  |  |
| --- | --- |
| Логічні умови | Позначення |
| RG2[n] | x1 |
| RG2 = 0 | x2 |

Закодований функціонально-структурний мікроалгоритм приведений на рисунку 2.4.



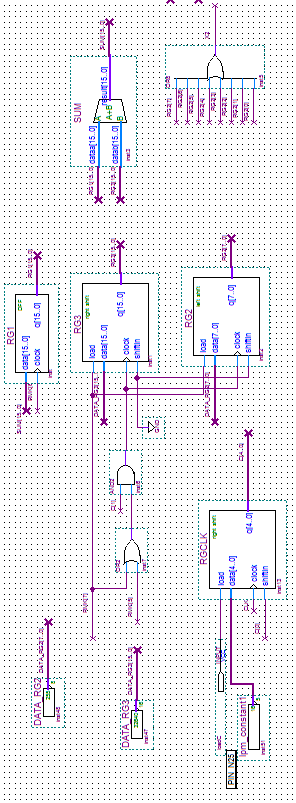
Рисунок 2.4− Закодований функціонально-структурний мікроалгоритм

2**.1.6 Опис роботи схеми АЛП**

Схема АЛП зображена на рисунку 2.5.

Часові діаграми роботи АЛП зображені на рисунку 2.6.

На схему АЛП подаються управляючі сигнали і синхросигнал, які управляють трьома регістрами: в один-дані тільки записуються, а два інші зсувають свій вміст або вліво, або вправо відповідно. Запис у зсуваючі регістри здійснюється, коли подається сигнал запису і сигнал зсуву одночасно, причому сигнал запису повинен приходити раніше, це забезпечується комбінаційною логікою: додані елементи «І». АЛП формує сигнали Z-нульовий вміст R2 і CX-молодший розряд R2.



|  |
| --- |
|  |

Рисунок 2.5-Схема АЛП

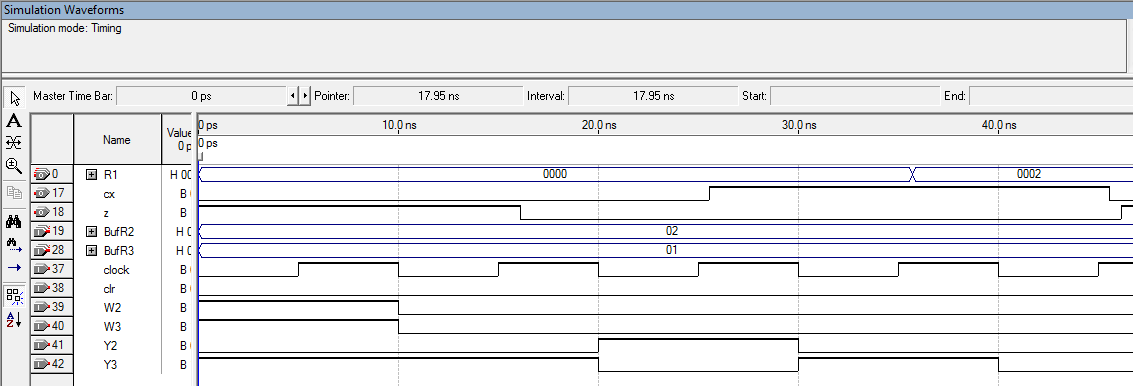


Рисунок 2.6-Часові діаграми роботи АЛП

**2.2 Опис БМУ**

**2.2.1 Розрахунок параметрів БМУ**

Визначимо параметри зони β1:



Визначимо розрядність адреси ПМК:



Розмістимо мікрокоманди в пам’яті, початкова адреса - 000000 (рисунок 2.7).

|  |  |
| --- | --- |
| Адреса | Команда |
| 00 | 1(begin) |
| 01 | 2 |
| 02 | 3 |
| 03 | 4 |
| 04 | 7(end) |
| 05 | 5 |
| 06 | 6 |

Рисунок 2.7−Розміщення команд у ПМК

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



За технічним завданням, тривалість мікрооперації підсумовування становить 4 такта. 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 = 4 – 1 =310 =112ПК= 1.012ДК.

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).

Таблиця 2.3−Таблиця кодування розрядів поля управління мультиплексором

|  |  |
| --- | --- |
|  | УС |
| 00 | 0 |
| 01 | Not(X1) |
| 10 | X2 |
| 11 | 1 |

Карта програмування БМУ зображена у таблиці 2.4, структурна схема БМУ зображена на рисунку 2.8.

Таблиця 2.4−Карта програмування БМУ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № МК | Адреса МК  . | β1 | | β2 | | | | β3 | | | | β4 | |
| S | M | Y1 | Y2 | Y3 | Y4 | SP |  | T1 | T2 | |  | |
| 1 | 000000 | 001 | 00 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | | 1 | |
| 2 | 000001 | 001 | 00 | 1 | 0 | 0 | 0 | 0 |  | 0 | 0 | | 0 | |
| 3 | 000010 | 001 | 00 | 0 | 1 | 0 | 0 | 0 |  | 0 | 0 | | 0 | |
| 4 | 000011 | 010 | 01 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | | 0 | |
| 5 | 000101 | 001 | 00 | 0 | 0 | 1 | 0 | 1 |  | 0 | 1 | | 0 | |
| 6 | 000110 | 101 | 10 | 0 | 0 | 0 | 1 | 0 |  | 0 | 0 | | 0 | |
| 7 | 000100 | 000 | 00 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | | 0 | |

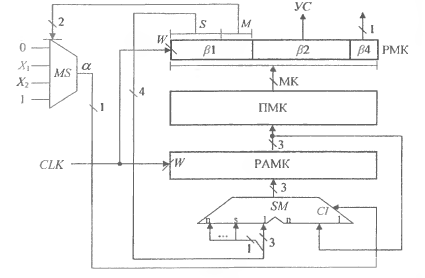


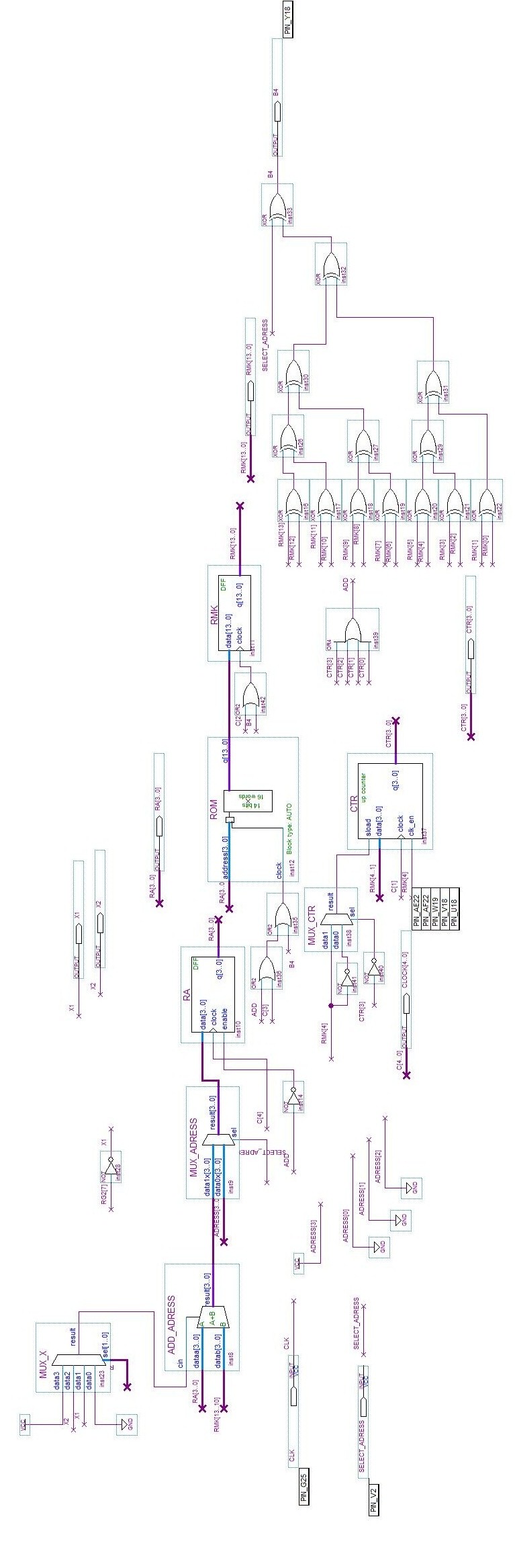
Рисунок 2.8−Структурна схема БМУ

**2.2.2 Опис роботи схеми БМУ.**

Схема БМУ зображена на рисунку 2.9.

Часові діаграми роботи БМУ зображені на рисунку 2.10.

Схема БМУ винонує управління обчислення, для цього вона формує управляючі сигнали, які подаються на АЛП. На АЛП також подається синхросигнал, який формується у БМУ шляхом поділу вхідної частоти на 4. БМУ також має входи Z і CX, які є виходами АЛП. БМУ містить регістр адреси, регістр команд, пам'ять команд, лічильник тактів, лічильник, для поділу частоти, суматор ,мультиплексом, для підключення відповідних сигналів і елементи комбінаційної логіки. Встановлення в нуль РА здійснюється через вхід clear. Частота ділиться для того, щоб команда встигала зчитатися з пам'яті. Запис у РА і виконання операції в АЛП відбувається по перепалу синхросигналу з 1 в 0, Запис у РК відбувається по перепаду синхросигналу з 0 в 1. Вхідний синхросигнал подається на пам'ять команд.



|  |
| --- |
|  |

Рисунок 2.9-Схема БМУ

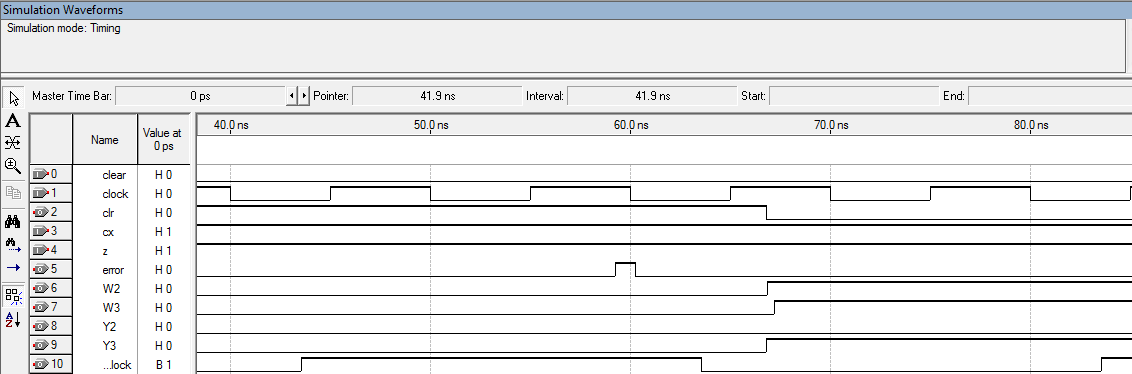


Рисунок 2.10-Часові діаграми роботи БМУ

**2.3 Опис операційного пристрою**

**2.3.1 Опис роботи операційного пристрою**

Схема пристрою зображена на рисунку 2.11

Часова діаграма роботи пристрою зображена на рисунку 2.12

Схема пристрою містить окремі схеми АЛП та БМУ і з'єднує їх. Має входи синхросигнала, встановлення адреси в нуль, входи задання операндів, і виходи помилки і результата обчислень.

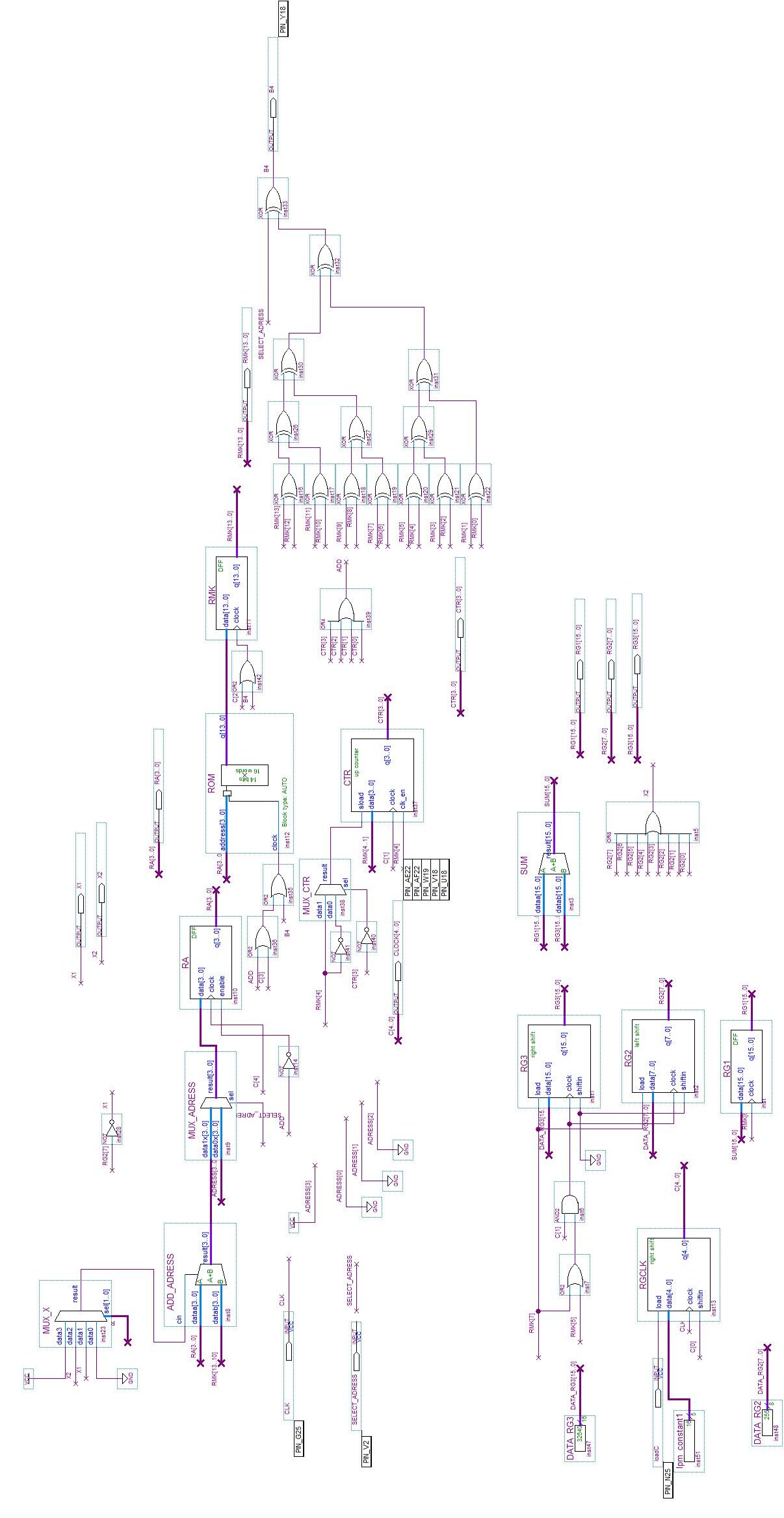


Рисунок 2.11-Схема розроблюваного пристрою

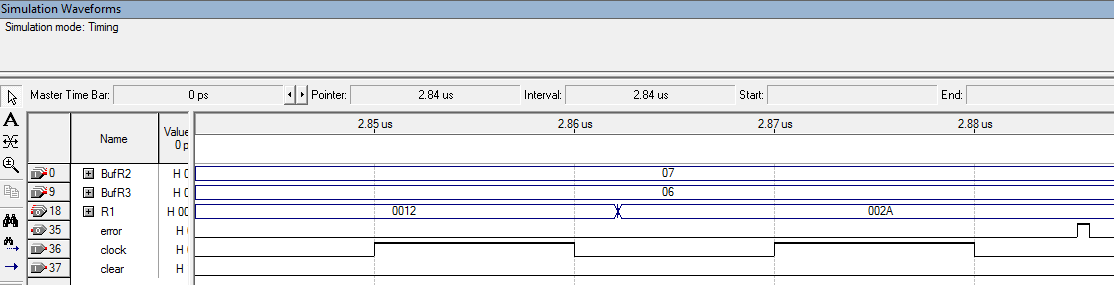


Рисунок 2.12-Часові діаграми роботи розроблюваного пристрою

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

**3.1 Загальні відомості**

Для набору, від лагодження та симуляції роботи створюваного пристрою використовувалася САПР Quartus || Version 9.1 Build 222 10/21/2009 SJFullVersion.Для проектування використовуємо сімейство пристроїв CycloneII, а з доступних пристроїв – EP2C35F672C6.

Необхідні елементи створюємо за допомогою мегафункцій(рис. 3.1).

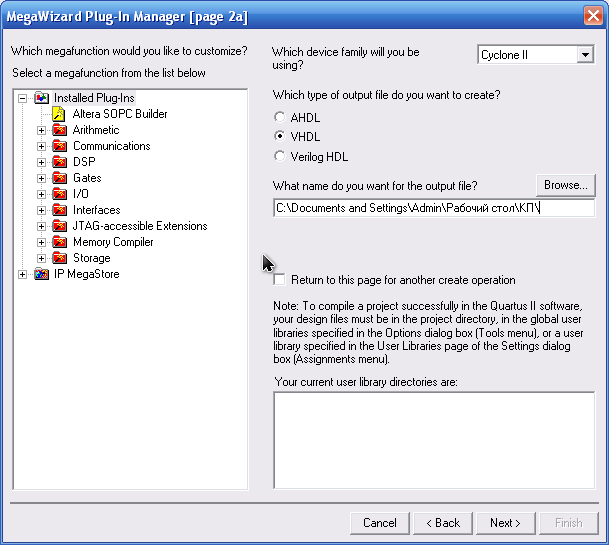


Рисунок 3.1-Вибір мегафункції-предка та імені створюваного елементу

Відповідність створюваних елементів та їх мегафункцій предків показана в табл.3.1

Таблиця 3.1 Мегафункції-предки елементів

|  |  |
| --- | --- |
| Елемент | Мегафункція-предок |
| Cуматор | LPM\_ADD\_SUB |
| Лічильник | LPM\_COUNTER |
| Регістр без зсуву | LPM\_FF |
| Мультиплексор | LPM\_MUX |
| Елемент збереження константи | LPM\_CONSTANT |
| Регістр із зсувом | LPM\_SHIFTREG |
| Пам’ять мікрокоманд | ROM: 1-PORT |
| Суматор за модулем 2 | LPM\_XOR |

При конфігурації пам’яті використаємо Memory Initialization File для вказання вмісту цієї памяті (рис.3.2).

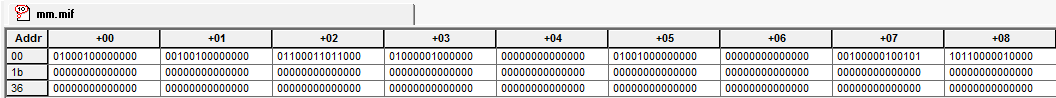


Рисунок 3.2-Створений файл вмісту пам’яті

**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).

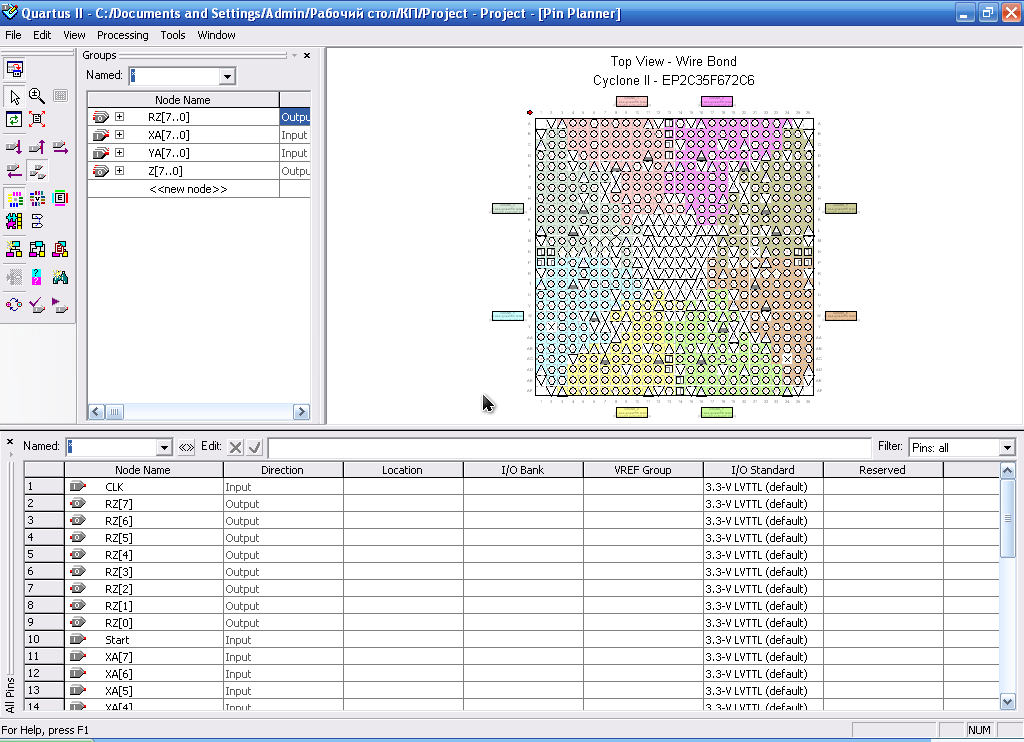


Рисунок 4.1-Вікно PinPlaner

Розпіновку отриманої схеми виконуємо відповідно до таблиці 4.1.

Таблиця 4.1 Розпіновка розроблюваного пристрою

|  |  |  |
| --- | --- | --- |
| clock | Input | PIN\_D13 |
| ERROR | Output | PIN\_AD12 |
| R1[15] | Output | PIN\_AE23 |
| R1[15] | Output | PIN\_AF23 |
| R1[15] | Output | PIN\_AB21 |
| R1[15] | Output | PIN\_AC22 |
| R1[15] | Output | PIN\_AD22 |
| R1[15] | Output | PIN\_AD23 |
| R1[15] | Output | PIN\_AD21 |
| R1[15] | Output | PIN\_AC21 |
| R1[15] | Output | PIN\_AA14 |
| R1[15] | Output | PIN\_Y13 |
| R1[15] | Output | PIN\_AA13 |
| R1[15] | Output | PIN\_AC14 |
| R1[15] | Output | PIN\_AD15 |
| R1[15] | Output | PIN\_AE15 |
| R1[15] | Output | PIN\_AF13 |
| R1[15] | Output | PIN\_AE13 |
| Clear | Input | PIN\_W1 |
| BufR2[7] | Input | PIN\_N25 |
| BufR2[6] | Input | PIN\_N26 |
| BufR2[5] | Input | PIN\_P25 |
| BufR2[4] | Input | PIN\_AE14 |
| BufR2[3] | Input | PIN\_AF14 |
| BufR2[2] | Input | PIN\_AD13 |
| BufR2[1] | Input | PIN\_AC13 |
| BufR2[0] | Input | PIN\_C13 |
| BufR3[7] | Input | PIN\_B13 |
| BufR3[6] | Input | PIN\_A13 |
| BufR3[5] | Input | PIN\_N1 |
| BufR3[4] | Input | PIN\_P1 |
| BufR3[3] | Input | PIN\_P2 |
| BufR3[2] | Input | PIN\_T7 |
| BufR3[1] | Input | PIN\_U3 |
| BufR3[0] | Input | PIN\_U4 |

Результат призначення зображено на рисунку 4.2.

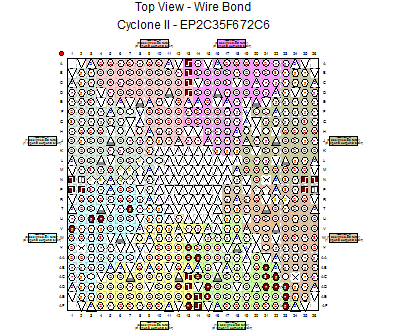


Рисунок 4.2-Результат призначення пінів

**4.2 Програмування ПЛІС**

Програмування ПЛІС проводиться за допомогою утиліти Programmer.

**5.** **Висновки**

В результаті виконання завдання на елементній базі FPGА був створений фрагмент процесорного ядра, що складається з арифметико-логічного пристрою 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 ||.

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.

1. **Список літератури:**
2. 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.
3. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. – Прикладна теорія цифрових автоматів: Навчальний посібник. – К.: Книжкове вид-во НАУ, 2007. – 364 с.
4. User Manual: Altera DE2 Board. - Altera Corporation, 2006. – 72 c.