

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЛП – арифметико-логічний пристрій
БМУ – блок мікропрограмного управління
БОД – блок обробки даних
БПП – блок пріоритетних переривань
ВП – віртуальний процесор
ВІС – велика інтегральна схема
ЕОМ – електронна обчислювальна машина
ІМС – інтегральна мікросхема
КС – контролер станів
КМП – контролер послідовності мікрокоманд
КПДП – контролер прямого доступу до пам'яті
МП – мікропроцесор
МС – мікросхема
МПК – мікропроцесорний комплект
МПП – мікропрограмна пам'ять
МПС – мікропроцесорна секція
ОС – обчислювальна система
ОЗП – оперативний запам'ятовувальний пристрій
ПВВ – пристрій вводу/виводу
ПЗП – постійний запам'ятовувальний пристрій
РЗП – регістровий запам'ятовувальний пристрій
СМУ – схема мікро програмного управління
СПП – схема прискореного переносу
СУСЗ – схема управління станами і зсувами
ТТЛ – транзисторно-транзисторна логіка
ФАМ – формувач адрес мікрокоманд
ЦПЕ – центральний процесорний елемент

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 2 |

ВСТУП

Обчислювальна техніка відіграє ключову роль у науково-технічному прогресі. Використання цифрових обчислювальних пристроїв дозволяє створювати на їх основі контрольні та керуючі системи, які інтегруються у прилади, машини, технологічні установки та процеси. Це сприяє значному підвищенню рівня автоматизації, економії енергії, сировини та матеріалів, а також покращенню продуктивності та якості праці.

Одним з важливих досягнень мікроелектроніки та обчислювальної техніки є розробка мікропроцесорів (МП). Використання мікропроцесорів у промисловому обладнанні дозволяє значно знизити їх вартість у порівнянні з системами на основі малих та середніх інтегральних схем, які виконують аналогічні функції. Одночасно з цим досягається суттєве зменшення маси, габаритів та енергоспоживання системи. Перехід на нову елементну базу підвищує технологічність систем промислової автоматики та значно розширює економічно доступну сферу їх застосування.

Промисловість освоїла та серійно випускає велику кількість мікропроцесорних комплектів великих інтегральних схем (ВІС) різної архітектури. Розвиток елементної бази мікропроцесорів здійснюється за кількома взаємодоповнюючими напрямками, що дозволяє гнучко адаптувати архітектуру систем управління до вимог конкретних застосувань та умов експлуатації. Останніми роками чітко простежується тенденція стандартизації систем команд та інтерфейсів електронно-обчислювальних машин (ЕОМ) та мікроконтролерів, що забезпечує взаємозамінність комп'ютерів та сумісність їхнього програмного забезпечення.

Мікропроцесор (МП) – це пристрій, який обробляє інформацію відповідно до програми, поданої командами на його входи, і реалізований

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 3 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

в одній або кількох великих інтегральних схемах (ВІС). Очевидно, що МП не може функціонувати без інших інтегральних мікросхем (ІМС), які виконують функції синхронізації, узгодження по навантаженню тощо.

Набір інтегральних схем, сумісних за конструктивно-технологічними характеристиками та призначених для створення мікропроцесорів, ЕОМ і мікропроцесорних систем, називається мікропроцесорним комплектом (МПК). МПК включає ВІС процесора, ВІС регістрів, ІМС регістрів, інтерфейсів пристроїв введення-виведення, ВІС контролерів, генераторів, таймерів тощо. Таким чином, мікропроцесор виконує арифметичні та логічні операції, аналізує та приймає рішення, які змінюють процес обчислень, керує процесом введення та виведення інформації, тобто виконує функції центрального процесора ЕОМ.

Для однокристальних 8-розрядних мікропроцесорів характерне використання регістрової та квазі-регістрової адресації, яка дозволяє в межах 8-розрядної команди задавати операції класу "регістр-регістр" або "регістр-пам'ять", що відповідає вимогам обробки програм керування. Однокристальні мікропроцесори мають фіксовану систему команд, не допускають розширення і мають універсальне призначення.

Секційні мікропроцесори мають довільну систему команд (реалізовану методом емуляції), допускають розширення, володіють вищою швидкістю та використовуються для створення специфічних інтегральних схем. При використанні секційних мікропроцесорів процесор ЕОМ будується на основі набору ВІС центральних процесорних елементів з блоком прискореного переносу, які утворюють операційну частину, та блоку мікропрограмного управління, який утворює пристрій керування.

Для реалізації інтерфейсу в склад секційних комплектів включаються ВІС магістральних приймачів-передавачів, контролерів переривань і контролерів інтерфейсів, а для генерації синхронізуючих

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 4 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

імпульсів процесора – контролерів синхронізації. Важливо розрізняти контролери та процесори команд, оскільки вони відрізняються кількістю рівнів управління. Мікроконтролери мають один рівень управління процесом обробки інформації – рівень мікрокоманд. Процесори команд мають два рівня управління процесом обробки інформації:

1. Рівень команд.
2. Рівень мікрокоманд.

Таким чином, процесори команд можуть виконувати практично довільну множину команд, маючи обмежену кількість мікрокоманд. Прикладний алгоритм зберігається у зовнішній пам'яті, тоді як в мікропрограмній пам'яті зберігаються мікропрограми емуляції команд. Замість виконання команди, виконується мікропрограма її емуляції.

Емуляція має такі переваги:

- спрощення структури мікропроцесора;
- можливість роботи з практично будь-якою системою команд;
- полегшення розробки мікропроцесора;
- можливе збільшення швидкодії завдяки використанню внутрішньої пам'яті.

Розроблювана мікроЕОМ призначена для реалізації системи команд. Вона створюється в навчальних цілях, щоб навчити проектувати системи обробки даних.

Мікропроцесорний комплект на основі однокристального процесора являє собою одну мікросхему, тоді як на основі секційного – кілька мікросхем, кожна з яких є мікропроцесорною секцією.

Структурно мікропроцесорна секція складається з таких частин:

- блок внутрішньої пам'яті;
- арифметико-логічний пристрій;
- блок реєстра Q;
- блок управління.

До секційних мікропроцесорних комплектів належить набір високошвидкісних великих інтегральних схем (ВІС) серії К1804, призначений для створення швидкодіючих мікропроцесорних пристроїв з розрядно-модульною організацією (ЕОМ середньої продуктивності, контролерів, засобів цифрової автоматики). Цей комплект призначений для реалізації структур машин з довільною системою команд. Секційна архітектура ВІС комплекту дозволяє нарощувати розрядність пристроїв, а управління здійснюється мікропрограмним способом. Орієнтація на обрану систему команд реалізується шляхом емуляції на мікропрограмному рівні, записуючи необхідні мікропрограми у постійний запам'ятовувальний пристрій. До серії К1804 входять наступні ВІС:

- Два типи мікропроцесорних секцій: К1804ВС1 та К1804ВС2.
- Два типи секційних блоків мікропрограмного управління: К1804ВУ1 і К1804ВУ2.
- Контролер послідовностей мікрокоманд: К1804ВУ3.
- Блок мікропрограмного управління: К1804ВУ4.
- Секційний контролер адреси: К1804ВУ5.
- Паралельний регістр: К1804ІР1.
- Блок прискореного переносу: К1804ВР1.
- Контролер станів: К1804ВР2.
- Три типи секційних магістральних приймачів-передавачів: К1804ВА1, К1804ВА2, К1804ВА3.
- Генератор синхронізації: К1804ГТ1.
- Багатоцільовий буферний регістр: К1804ІР2.
- Секційний порт вводу/виводу: К1804ІР3.

Усі ВІС працюють при напрузі живлення $5\text{В} \pm 0.25\text{В}$ та сумісні зі стандартними ТТЛ схемами.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 6 |

У курсовому проекті буде реалізовано процесор команд. Керуюча частина МП (Рис. 1.1) побудована на основі принципу мікропрограмного управління.



Рис. 1.1 Структура керуючої частини МП

Мікропрограма управління процесом або об’єктом знаходиться у мікропрограмній пам'яті (МПП). Формувач адрес мікрокоманд (ФАМ) генерує адреси мікрокоманд. У кожному такті поточна мікрокоманда зберігається в регістрі мікрокоманд. У наступних розділах буде проведений детальний опис блоків, що проектуються за варіантом курсового проекту.

РОЗДІЛ 1. СТРУКТУРА ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ З МІКРОПРОГРАМНИМ УПРАВЛІННЯМ

До складу обчислювальної системи з мікропрограмним управлінням входять наступні компоненти:

- блок обробки даних;
- блок мікропрограмного управління;
- блок пріоритетних переривань;
- пам'ять;
- зовнішні пристрої;
- контролер прямого доступу до пам'яті.

Характеристика блоку обробки даних:

До складу блоку обробки даних (БОД) входять арифметико-логічний пристрій (АЛП) та схема управління станом і зсувами (СУСЗ). АЛП забезпечує виконання арифметичних та логічних операцій і будується на основі мікропроцесорних секцій (МПС), кожна з яких обробляє дані в 4 розряди. Структурна схема МПС включає чотири блоки: блок внутрішньої пам'яті, блок операційного автомата (ОА), блок регістра Q та блок управління. Обчислювальна система є 32-розрядною, тому АЛП складається з восьми МПС K1804BC1.

Мікросхеми K1804BC1 мають наступні виводи (див. рис. 1.2 та рис. 1.3):

- МІ (9 розрядів) – визначає, яку саме мікрооперацію виконуватиме операційний автомат у поточному циклі;
- А, В (по 4 розряди) – адресні входи, дані надходять з регістрів А і В відповідно;
- D (4 розряди) – вхід з шини даних;
- Y (4 розряди) – вихід на шину Y;
- ОЕ – вхід для сигналу дозволу видачі результату на шину Y;

- C4 – перенос за межі старшого розряду результату;
- F3 – значення (копія) старшого розряду результату (знак результату);
- Z – ознака нульового результату;
- OVR – переповнення результату;
- P – сигнал поширення переносу;
- G – сигнал генерації переносу;
- CI – вхідний перенос (перенос у нульовий розряд);
- PQ3 – вивід зсуву старшого розряду регістра Q;
- PQ0 – вивід зсуву молодшого розряду регістра Q;
- PF0 - PF3 – виводи, які залежно від напрямку зсуву є входом або виходом, через які здійснюється запис значення в розряд, що звільняється, і видача вмісту розряду, що виштовхується.

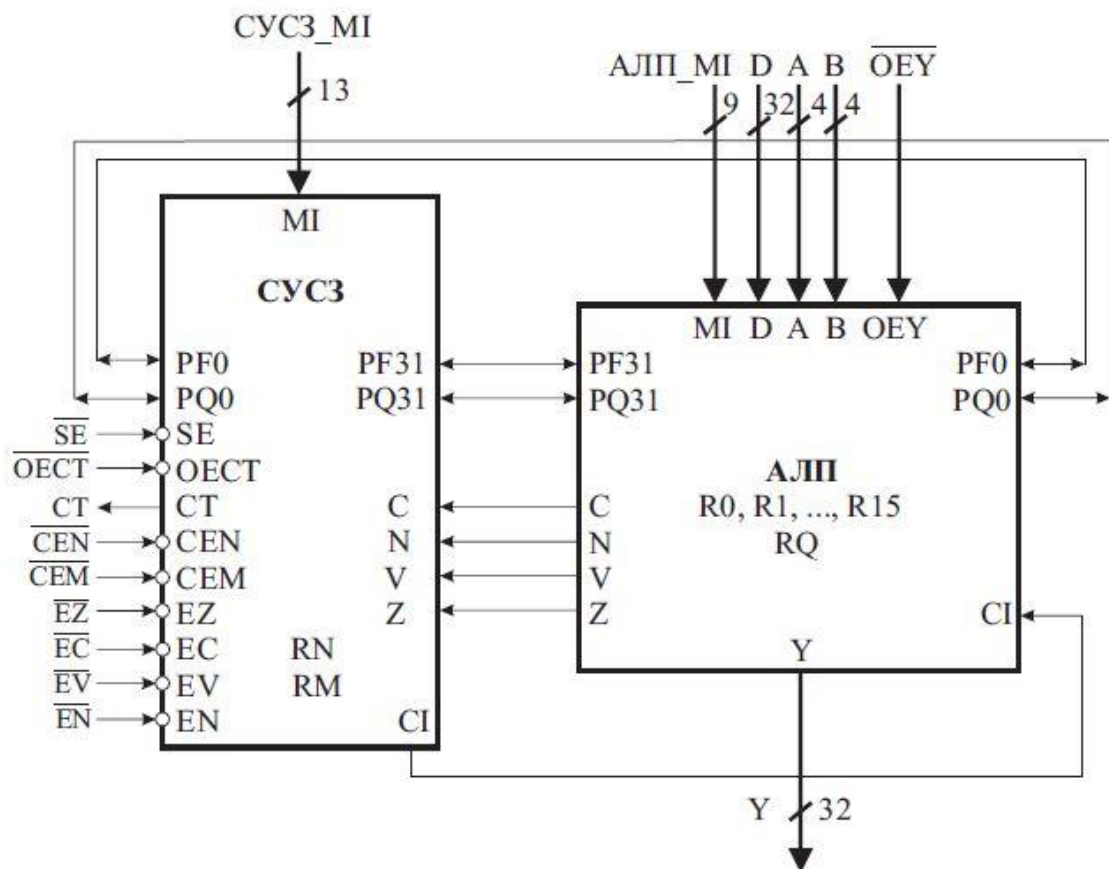


Рис. 1.2 Узагальнена структура блока обробки даних

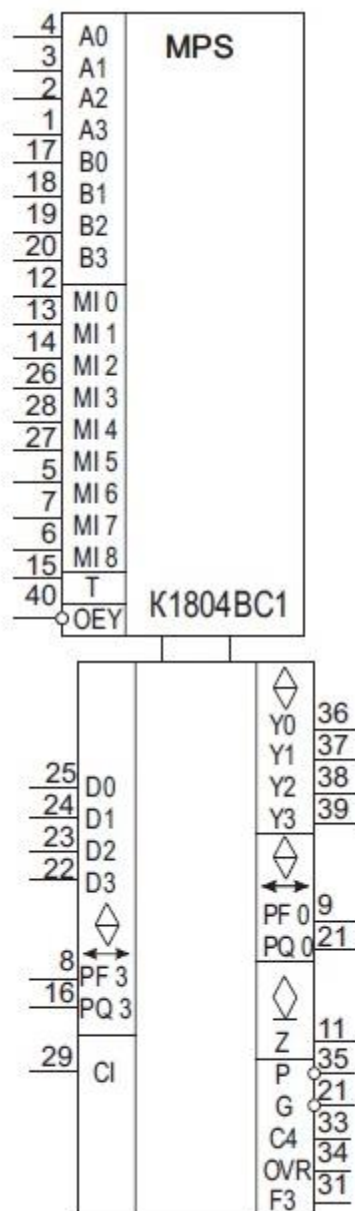


Рис. 1.3 Умовне графічне позначення схеми МПС К1804ВС1 на
принципових схемах

Схема управління станами і зсувами (мікросхема К1804ВР2)
призначена для наступних завдань:

- Формування сигналу вхідного переносу (CI) для молодшої МПС АЛП та СПП.
- Забезпечення арифметичних, логічних, циклічних та інших зсувів даних в АЛП.
- Виконання мікрооперацій над бітами слова стану (C4, F3, Z, OVR).

- Формування сигналу умови (CondiTion).

Призначення виводів мікросхеми K1804BP2 (див. рис. 1.4):

- MI0 – MI12 – входи для мікроінструкцій.
- C, N, V, Z – входи слова стану.
- YC, YN, YV, YZ – входи двонаправленої шини Y для запису слова стану у зовнішній регістр та читання з регістра.
- OEY – дозвіл на видачу слова стану на шину Y.
- CEN, CEM – дозвіл запису в регістри N і M відповідно.
- EZ, EN, EV, EC – входи керування записом в окремі розряди регістру M.
- CT – вихід сигналу умови (тристабільний).
- OECT – дозвіл видачі сигналу умови CT на шину.
- PF3, PF0, PQ3, PQ0 – входи/виходи для організації зсувів.
- SE – сигнал дозволу зсуву.

Після виконання кожної мікрооперації формується слово стану, яке складається з наступних ознак результату:

- C4 – перенос за межі старшого розряду результату.
- F3 – значення (копія) старшого розряду результату (знак результату).
- Z – ознака нульового результату (вихід з відкритим колектором).
- OVR – ознака переповнення результату.

Після виконання кожної мікрооперації в АЛП зі старшої МПС в СУСЗ надходять ознаки результату C4, F3, OVR, Z, де над ними можуть виконуватись різноманітні операції. Поточні або попередні значення слова стану можуть зберігатись у двох внутрішніх регістрах (RgN, RgM) або у зовнішньому стеку. На підставі слова стану формується сигнал

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 11 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

умови СТ, який надходить в блок мікропрограмного управління й використовується для організації умовних переходів в мікропрограмах.

Два можливих способи об'єднання мікропроцесорних секцій в АЛП:

- З послідовним переносом із секції в секцію.
- З паралельним переносом.

У даній роботі використовується спосіб об'єднання схем з паралельним переносом із секції в секцію. В такому випадку для об'єднання мікропроцесорних секцій використовується схема прискореного переносу (див. Рис. 1.5).

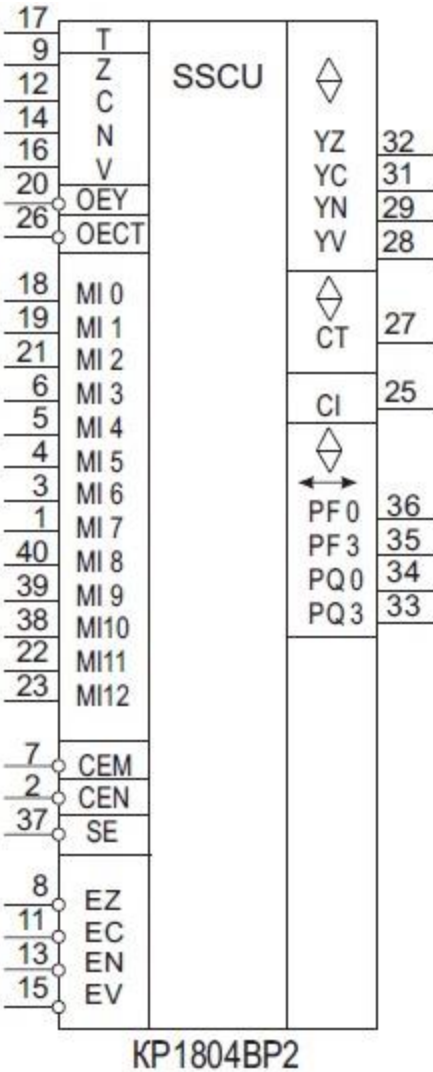


Рис. 1.4 Умовне графічне позначення схеми СУСЗ К1804ВР2 на принципових схемах

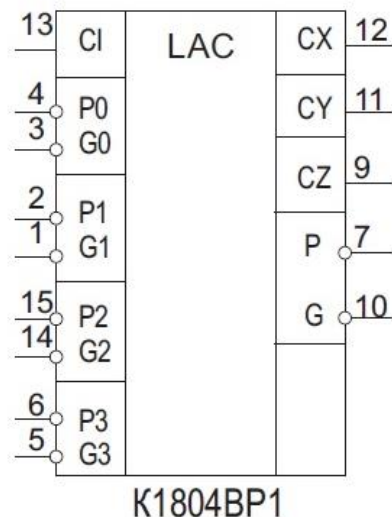


Рис. 1.5 Умовне графічне позначення схеми прискореного переносу
K1804BP1 на принципових схемах

Обов'язкові компоненти блоку мікропрограмного управління (БМУ,
Рис. 1.6):

1. ФАМ (формувавч адреси мікрокоманд):

- Використовується спеціальна мікросхема K1804BY4.
- Генерує 12-розрядні адреси мікрокоманд.
- Виконує 16 мікроінструкцій, включаючи отримання наступної адреси, повторення адреси, умовні та безумовні переходи, виклики мікропідпрограм та організацію циклів.
- Виводи:
 - D0 – D11: шина адреси наступної команди.
 - RLD: дозвіл запису до регістру адреси/лічильника циклу.
 - CI: вхідний перенос лічильника мікрокоманд.
 - FL: ознака переповнення внутрішнього стеку (глибина стеку – п'ять 12-розрядних мікрокоманд).
 - A0 – A11: шина адреси мікрокоманди.
 - CC: код умови.
 - CCE: код дозволу умови.

- Функції: формує адресу мікрокоманди, використовується для організації мікропрограм.
2. ПМК (пам'ять мікрокоманд):
 - Зберігає мікропрограму емуляції мікрокоманд та мікропрограми обслуговування зовнішніх пристроїв.
 3. РМК (регістр мікрокоманд):
 - Зберігає поточну мікрокоманду, що виконується.
 4. МУ (мультиплексор умови):
 - Використовується для вибору умови для організації умовних переходів.
 5. ІНВ (інвертор умови):
 - Являє собою суматор за модулем 2, необхідний для виконання умовних переходів.

Допоміжні компоненти:

- ППА (перетворювач початкової адреси): перетворює код операції мікрокоманди в початкову адресу мікропрограми.
- ПА (перетворювач адреси): трансформує вектор переривання в початкову адресу мікропрограми обслуговування даного переривання.

Мікросхема К1804ВУ4 формує 3 сигнали управління:

- RE: дозвіл РМК.
- ME: дозвіл перетворення початкової адреси (ППА).
- VE: дозвіл перетворення адреси.

Ці компоненти забезпечують основні функції для керування мікропрограмами та взаємодії з зовнішніми пристроями в обчислювальній системі з мікропрограмним управлінням.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 14 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

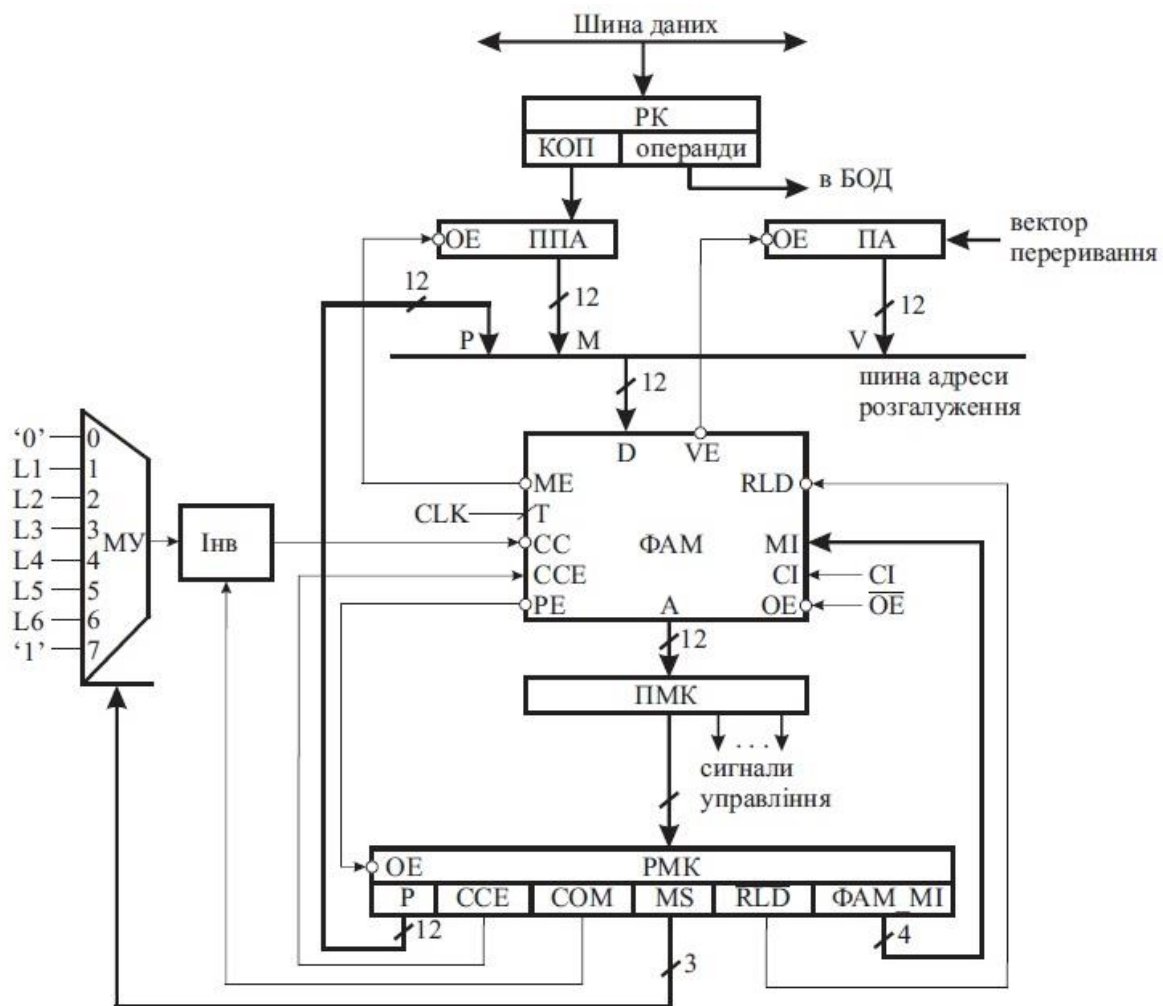


Рис. 1.6 Структура блока мікропрограмного управління

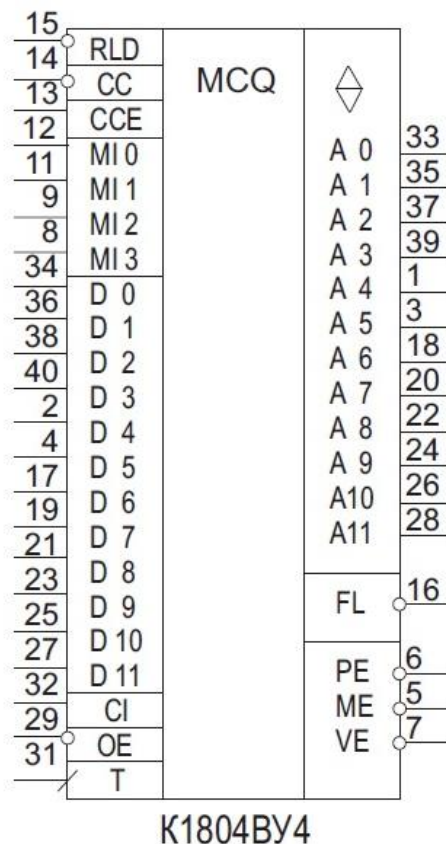


Рис. 1.7 Умовне графічне позначення мікросхеми формувача адрес мікрокоманд K1804BY4 на принципових схемах

Мікросхема K1804BH1 призначена для прийому та обслуговування запитів на переривання від 8 зовнішніх пристроїв. Вона формує вектор переривання і має наступні виводи:

- INR0 – INR7: входи запитів на переривання.
- ER: визначає форму сигналів запитів.
- MK0 – MK7: шина маски.
- S0 – S2: шина стану.
- MI0 – MI3: входи мікроінструкцій.
- EI: дозвіл виконання мікроінструкцій.
- OVR: сигнал переповнення.
- VEC0 – VEC2: вектор переривання.

Чотири розряди коду мікроінструкції МІ0–МІ3 дозволяють створити систему команд БПП з 16 команд:

- RESET IR: очистка регістра запитів.
- CLR IR, val: очистка окремих розрядів регістра запитів за значеннями з шини маски.
- CLR IR, MR: очистка окремих розрядів регістра запитів за значеннями з регістра маски.
- CLR IR, VR: очистка одного з розрядів регістра запитів під управлінням вектора запиту.
- LOAD MR, val: завантаження регістра маски значеннями val.
- READ MR: читання регістра маски.
- CLR MR, val: очистка регістра маски за значеннями з шини маски.
- SET MR, val: встановлення в «1» окремих розрядів регістра маски за значеннями з шини маски.
- SET MR: встановлення в «1» всіх розрядів регістра маски.
- RESET MR: очистка регістра маски.
- LOAD SR, val: завантаження регістра стану значеннями val.
- READ SR: читання слова стану.
- READ VR: читання вектора переривань.
- EI: дозвіл переривань.
- DI: заборона переривань (виходу з регістра запитів IR).
- RESET: загальна очистка схеми.

Мікросхема К1804ВР3 дозволяє об'єднати до 8 мікросхем К1804ВН1 для створення блоку пріоритетних переривань (БПП) з кількістю ліній переривань до 64. Вона має наступні виводи:

- INR0 – INR7: входи запитів на переривання.
- ER: визначає форму сигналів запитів.
- MK0 – MK7: шина маски.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 17 |

- S0 – S2: шина стану.
- MI0 – MI3: входи мікроінструкцій.
- EI: дозвіл виконання мікроінструкцій.
- OVR: сигнал переповнення.
- VEC0 – VEC2: вектор переривання.

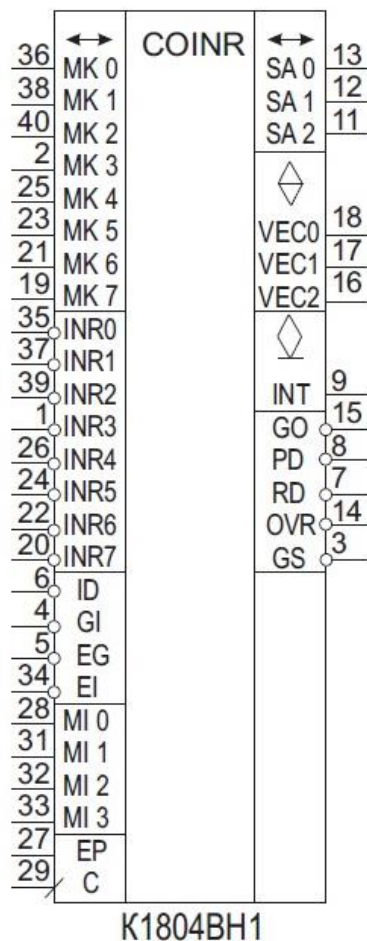


Рис. 1.8 Умовне графічне позначення мікросхеми векторних переривань
K1804BH1 на принципових схемах

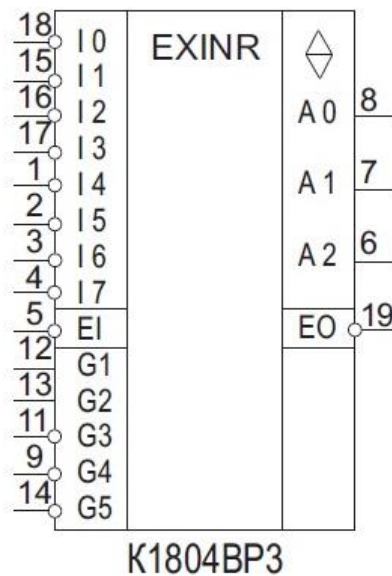


Рис. 1.9 Умовне графічне позначення мікросхеми розширювача векторних переривань К1804BP3 на принципових схемах

Прямий доступ до пам'яті (ПДП) – спосіб вводу/виводу інформації, під час якого встановлюється безпосередній зв'язок між приладами вводу/виводу (ПВВ) і пам'яттю, а передача даних виконується без участі центрального процесору. Досягається висока швидкодія обміну інформацією, звільнення центрального процесору від безпосередньої передачі даних. Для організації ПДП використовують контролер ПДП (КПДП).

Функції КПДП:

- керування режимами передачі;
- генерація адреси;
- пересилка даних;
- підрахунок кількості слів.

Основною структурною одиницею КПДП є генератор адреси (Г А) К1804ВУ6.

Функції мікросхеми К1804ВУ6:

- формування послідовних адрес комірок пам'яті під час передачі даних до пам'яті або з пам'яті;

- підрахунок кількості слів;
- формування сигналу кінця передачі.

Для встановлення зв'язку між ПВВ і пам'яттю використовують механізм пріоритетних переривань. БПП приймає запит на переривання від ПВВ, формує сигнал IR – вимога переривання і вектор переривання ПА трансформує вектор переривання в 12-розрядну початкову адресу мікропрограми обслуговування ПВВ, яка знаходиться в МПП. Ця програма налаштовує ГА на потрібний режим роботи, задає кількість слів, початкову або кінцеву адресу пам'яті та ініціює початок передачі інформації. Дані з ПВВ передаються через внутрішню три-стабільну шину на шину даних пам'яті (через приймально-передавальний пристрій), а адресу формує ГА.

Система команд ГА:

- 000 WRCR – запис в регістр управління
- 001 RDCR – читання з регістра управління
- 101 LDAD – завантаження регістра та лічильника адреси
- 110 LDWC – завантаження регістра кількості слів та лічильника слів
- 011 RDAC – читання лічильника слів
- 010 RDWC – читання лічильника слів
- 100 REIN – завантаження лічильників з регістрів
- 111 ENCT – запуск (дозвіл лічби)

Режими роботи K1804BY6:

- Режим 0: До появи '0' у лічильнику слів

До регістру керування заноситься 100, до регістру кількості слів і лічильника слів – кількість слів, до регістру і лічильника адреси – кінцева адреса та відбувається зменшення значень лічильників на 1 на кожному кроці до появи у лічильнику слів '0'.

- Режим 1: Передача до заданої кількості слів

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 20 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

До регістру управління заноситься 001, до регістру кількості слів – кількість слів, до лічильника слів – 0, до регістру і лічильника адреси – початкова адреса та відбувається збільшення значень лічильників на 1 на кожному кроці до появи рівності регістру та лічильника слів.

- Режим 2: Передача до заданої адреси

В цьому режимі виконується передача до заданої адреси, причому лічильник слів виконує функції регістра. Операція пересилки зупиняється, коли вміст лічильника адреси дорівнює вмісту лічильника слів. До регістру управління заноситься 010, до регістру слів – кінцева адреса, до лічильника адреси – початкова адреса та відбувається збільшення значення лічильника адреси на 1 при кожному кроці до появи рівності лічильника слів та адреси.

- Режим 3: Передача до переповнення лічильника слів

До лічильника слів записується доповнення кількості слів, які необхідно передати. Лічильник слів виконує інкремент до переповнення (сигнал WCO). До регістру управління заноситься 011, до лічильника і регістра слів – кількість слів в доповняльному коді, до лічильника і регістра адреси – початкова адреса.

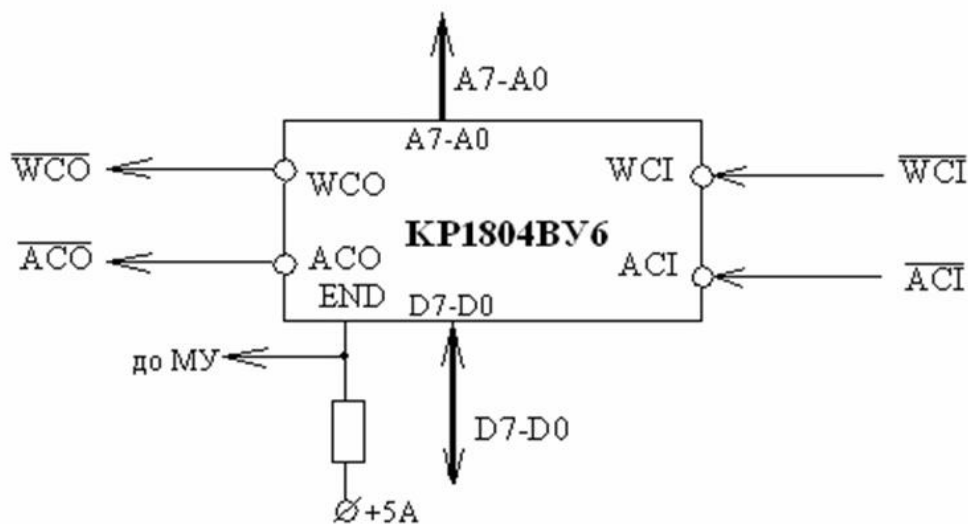


Рис. 1.10 Умовне графічне позначення контролера прямого доступу до пам'яті К1804ВУ6

РОЗДІЛ 2. ОСНОВНА ПАМ'ЯТЬ ОБЧИСЛЮВАЛЬНОЇ СИСТЕМИ

Розробка функціональної схеми основної пам'яті ЕОМ включає в себе кілька ключових етапів. По-перше, потрібно визначити потрібну ємність оперативної та постійної пам'яті відповідно до вимог системи. Далі необхідно визначити кількість субмодулів мікросхем пам'яті, які потрібні для досягнення необхідної розрядності шини даних. Важливо також розподілити адресний простір між оперативною та постійною пам'яттю, враховуючи потреби системи.

Необхідно спроектувати функціональну схему основної пам'яті ЕОМ ємністю 50 Кбайт.

Основна пам'ять складається з:

- ОЗП, ємністю 32 Кб;
- ПЗП, ємністю 16 Кб.
- Елементна база ОЗП: К541РУ2.
- Елементна база ПЗП: К555РЕ4.

К541РУ2. Мікросхема має 10 адресних входів, а також 4-розрядну двонапрямну тристабільну шину для вводу/виводу інформації і дозволяє зберігати 1 Кх4 біт інформації.

Керуючі сигнали:

- CS – вибір мікросхеми;
- OE – дозвіл видачі даних на шину;
- W/R – запис/читання.

К555РЕ4. Мікросхема має 11 адресних входів та 8-розрядну шину виводу і дозволяє зберігати 2 Кх8 біт інформації.

Керуючі сигнали:

- CS3, CS2, CS1 – для вибору мікросхеми.

Оскільки мінімальна одиниця інформації, що адресується – 32-розрядне слово, то ємність основної пам'яті складає 10К 32-розрядних слів.

- ОЗП – 8К 32-розрядних слів.
- ПЗП – 2К 32-розрядних слів.

Розрядність шини адрес для пам'яті:

- ОП = $\lceil \log_2 10K \rceil = \lceil \log_2 10 \rceil + 10 = 14$

Для досягнення необхідної розрядності МС пам'яті об'єднуються в субмодулі. Кількість мікросхем ОЗП у субмодулі – 8, ПЗП – 4.

Для досягнення необхідної ємності субмодулі об'єднуються в модулі. Кількість субмодулів ОЗП у модулі – 8, ПЗП – 2.

Таблиця 2.1 Розподіл адресного простору

| | | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-------|-------|-----|-----|-----|-----|-----|-------|
| | A14 | A13 | A12 | A11 | A10 | A9 | ... | ... | ... | ... | ... | A0 | HEX |
| ОЗП | 0 | 0 | 0 | 0 | 0 | 0 | ... | ... | ... | ... | ... | 0 | 0000h |
| | ... | ... | DC | | | ... | P Y 2 | | | | | ... | ... |
| | 0 | 0 | 1 | 1 | 1 | 1 | ... | ... | ... | ... | ... | 1 | 1FFFh |
| ПЗП | 0 | 1 | 0 | 0 | 0 | 0 | ... | ... | ... | ... | ... | 0 | 2000h |
| | ... | DC | | | ... | P E 4 | | | | | | ... | ... |
| | 1 | 1 | 1 | 1 | 1 | 1 | ... | ... | ... | ... | ... | 1 | 7FFFh |

Розряд A14 використовується для вибору ОЗП або ПЗП.

Розряди A9 – A0 подаються на мікросхеми оперативної пам'яті.

Розряди A10 – A0 подаються на мікросхеми постійної пам'яті.

Розряди A12 – A10 подаються на дешифратор ОЗП для вибору субмодулів оперативної пам'яті.

Розряди A13 – A11 подаються на дешифратор ПЗП для вибору субмодулів постійної пам'яті.

РОЗДІЛ 3. ОСОБЛИВОСТІ ПОБУДОВИ БЛОКА МІКРОПРОГРАМНОГО УПРАВЛІННЯ ТА БЛОКА ПРІОРИТЕТНИХ ПЕРЕРИВАНЬ

Блок мікропрограмного управління:

Основний елемент блока мікропрограмного управління ФАМ побудований на основі елементної бази мікросхеми K1804BY4 (DD1). Сигнал СС потрапляє в ФАМ з інвертора умови (DD2.1), який реалізовано на мікросхемі K555ЛЛЗ. На інвертор умови подається сигнал СОМ (РМК6) і сигнал з мультиплексора умови. Мультиплексор умови (DD4) реалізовано на мікросхемах КР1533КП15. МПП складається з двох субмодулів мікросхем КР556РТ7 (DD5 – DD15 та DD16 – DD26). Вибір субмодуля відбувається по сигналу ФАМ ОЕ та старшому розряду адреси А11 (DD3.1 використовується для інвертування сигналу А11, щоб вибрати перший субмодуль МПП). Субмодулі складаються з 11 мікросхем кожен та забезпечують необхідну розрядність МПП (84 розряди). Виходи 1 – 84 МПП подаються на РМК – регістр мікрокоманд, який реалізовано на основі елементної бази мікросхем K580ИР82 (DD27 – DD37).

Блок пріоритетних переривань:

Обчислювальна система з мікропрограмним управлінням розрахована на 16 зовнішніх пристроїв, тому БПП складається з двох схем векторних переривань K1804BH1 (DD38, DD39).

Розширювач векторних переривань (РВП) використовується для об'єднання декількох схем векторних переривань, щоб таким чином можна було під'єднати до обчислювальної системи більшу кількість зовнішніх пристроїв. Можливі два способи об'єднання схем векторних переривань: послідовний та паралельний. За варіантом реалізовано послідовне підключення.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 24 |

РВП реалізовано на МС К1804ВРЗ (DD40). На входи І РВП під'єднуються виходи GS СВП, на входи G подається мікроінструкція, вхід EI відповідає за дозвіл запису. На виході A0 матимемо останній розряд адреси вектора переривання VEC. Даний сигнал подаємо на ПА, реалізований на двох мікросхемах КР556РТ7 (DD41, DD42), який генерує адресу мікропрограми обробки даного переривання та подає її на вхід ФАМ.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 25 |

РОЗДІЛ 4. ФОРМАТИ ДАНИХ

32-розрядна обчислювальна система з мікропрограмним управлінням оперує з такими форматами даних:

- ### 1. Цілі 8-бітні числа без знаку (формат CHAR)

7

0

| |
|--|
| |
|--|

Діапазон: $0 \dots 2^8 - 1$

- ## 2. Цілі 32-бітні числа без знаку (формат INT)

31

0

| |
|--|
| |
|--|

Діапазон: $0 \dots 2^{32} - 1$

- ### 3. Цілі 32-бітні числа зі знаком (формат INTG)

31

30

0

| | |
|------|--|
| ЗНАК | |
|------|--|

Діапазон: $-2^{31} \dots 2^{31}-1$

4. Числа з плаваючою крапкою (формат REAL)

31

30

23

22

0

| знак | характеристика числа | мантиса числа |
|------|----------------------|---------------|
|------|----------------------|---------------|

Діапазон: $1.17549 \times 10^{-38} \dots 3.402823 \times 10^{38}$

- ## 5. Десяткові числа (формат VCD)

31

0

| | | | | | | | |
|--|--|--|--|--|--|--|--|
| | | | | | | | |
|--|--|--|--|--|--|--|--|

Діапазон: 0 ... 99 999 999

РОЗДІЛ 5. ПРОЕКТУВАННЯ СИСТЕМИ КОМАНД

5.1 Емуляція команд

ЕОМ здійснює емуляцію системи команд віртуального процесора. Це означає, що кожна команда, яка реалізована в процесорі емуляції, замінюється в ЕОМ послідовністю мікрокоманд (мікропрограмою) з системи команд, реалізованої в апаратній логіці ЕОМ.

Програмістська модель віртуального процесора включає 4 компоненти:

- 10 регістрів загального призначення (REGa ... REGj);
- програмний лічильник віртуального процесора (PC);
- слово стану віртуального процесора (PSW);
- показчик стеку (SP).

Компоненти віртуального процесора розміщуються у молодших адресах віртуальної пам'яті:

| Адреса | ОЗП |
|--------|----------|
| ... | Програма |
| ... | |
| 0Ch | SP |
| 0Bh | PSW |
| 0Ah | Rj |
| 09h | Ri |
| 08h | Rh |
| 07h | Rg |
| 06h | Rf |
| 05h | Re |
| 04h | Rd |
| 03h | Rc |
| 02h | Rb |
| 01h | Ra |
| 00h | PC |

Структура PSW:

| | | | | | | | | | | | | | | | |
|----|----|--|----|--|----|--|---|--|---|--|---|--|---|--|---|
| 15 | 14 | | 12 | | 10 | | 8 | | 6 | | 4 | | 2 | | 0 |
| | | | | | O | | S | | Z | | A | | P | | C |

O (Overflow) –ознака переповнення;

S (Sign) – ознака від'ємного результату;

Z (Zero) – ознака нульового результату;

A (Auxiliary carry) – ознака переносу з молодшої тетради;

P (Parity) – ознака парності;

C (Carry) – ознака переносу в старший розряд.

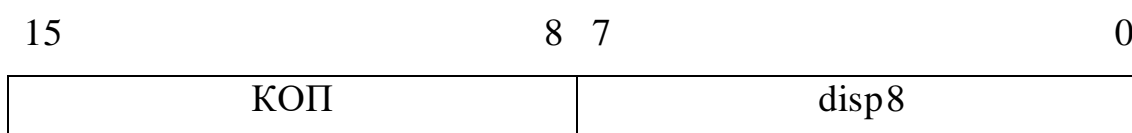
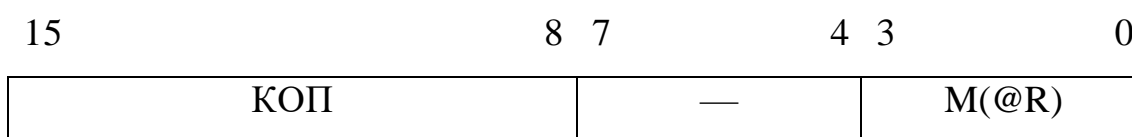
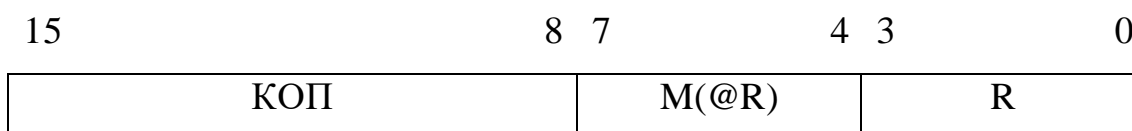
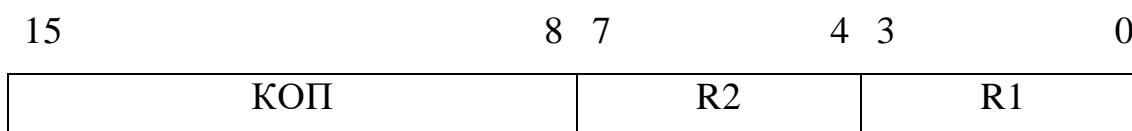
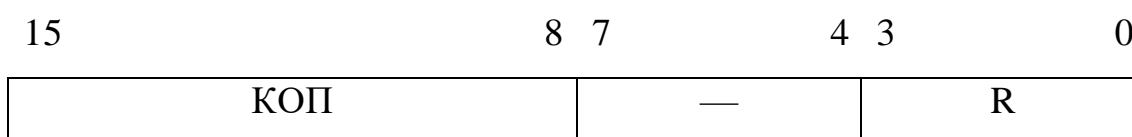
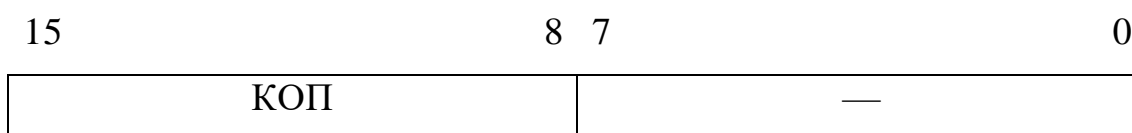
Узагальнений алгоритм емуляції:

1. Читання регістра програмного лічильника віртуального процесора з ОЗП в один з регістрів БОД.
2. Формування адреси команди.
3. Читання команди з ОЗП.
4. Виконання команди:
 - перехід на мікропрограму емуляції;
 - виконання мікропрограми.
5. Запис результату в ОЗП.
6. Модифікація регістра слова стану:
 - читання слова стану з ОЗП в регістр БОД;
 - модифікація значень ознак;
 - запис слова стану в ОЗП;
 - модифікація програмного лічильника;
 - запис нового значення лічильника в ОЗП.

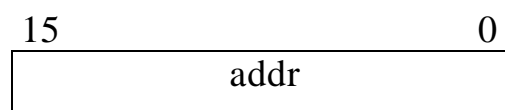
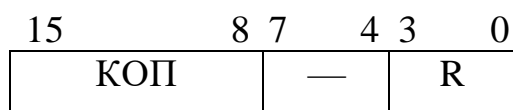
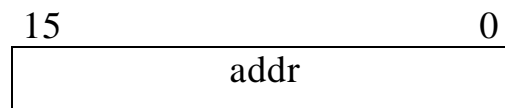
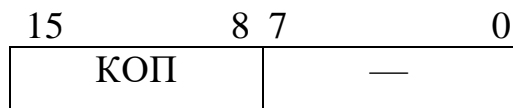
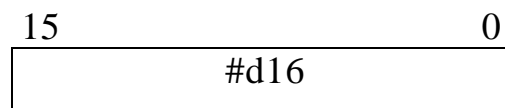
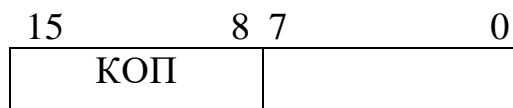
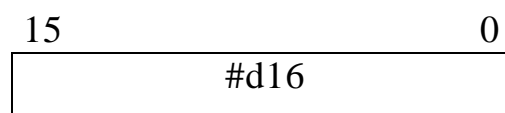
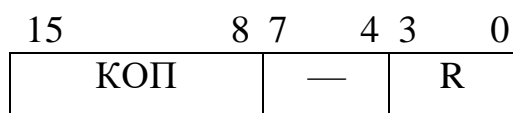
5.2 Формати команд

- КОП (Operation Code) – код операції
- R, R1, R2 – регістри (RA ... RJ)
- M – непряма адресація пам'яті (адреса комірки пам'яті у регістрі)
- addr – 16-розрядна адреса комірки пам'яті
- disp8 – зміщення (1 байт). Діапазон: -128 ... 127
- #d16 – безпосередні дані
- SP – регістр покажчика стеку (Stack Pointer)
- PC – лічильник команд (Program Counter)

Двобайтні команди:



Чотирибайтні команди:



5.3 Система команд

Команди пересилки даних:

```

mov      R,M;
mov      M,R;
mov      R1,R2;
mov      R,#d16;
xchg     R1,R2;
xchg     R,M;
push     R;
pop      R;

```

Арифметичні команди:

| | | | |
|------|--------|------|---------|
| add | R1,R2; | add | R,M; |
| add | M,R; | add | R,#d16; |
| adc | R1,R2; | adc | R,M; |
| adc | M,R; | adc | R,#d16; |
| addf | R1,R2; | addf | R,M; |
| addf | M,R; | addf | R,#d16; |
| addb | R1,R2; | inc | R; |
| sub | R1,R2; | sub | R,M; |
| sub | M,R; | sub | R,#d16; |
| subc | R1,R2; | subc | R,M; |
| subc | M,R; | subc | R,#d16; |
| subf | R1,R2; | subf | R,M; |

| | | | |
|------|---------|------|---------|
| subf | M,R; | subf | R,#d16; |
| subb | R1,R2; | dec | R; |
| neg | R; | mul | R1,R2; |
| mul | R,M; | mul | M,R; |
| mul | R,#d16; | mulf | R1,R2; |
| mulf | R,M; | mulf | M,R; |
| mulf | R,#d16; | div | R1,R2; |
| div | R,M; | div | M,R; |
| div | R,#d16; | divf | R1,R2; |
| divf | R,M; | divf | M,R; |
| divf | R,#d16; | sqr | R; |
| sqr | R; | pow | R1,R2; |
| not | R; | and | R1,R2; |
| or | R1,R2; | xor | R1,R2; |
| nxor | R1,R2; | nand | R1,R2; |
| srl | R1,R2; | sll | R1,R2; |
| sra | R1,R2; | sla | R1,R2; |
| src | R1,R2; | slc | R1,R2; |

Команди порівняння:

| | | | |
|-----|--------|-----|-----|
| cmp | R1,R2; | cmp | R,M |
| cmp | R,#d16 | | |

Команди переходів:

| | |
|------|-------|
| jmp | addr |
| ja | disp8 |
| jna | disp8 |
| jc | disp8 |
| jnc | disp8 |
| jz | disp8 |
| jnz | disp8 |
| jo | disp8 |
| jno | disp8 |
| jp | disp8 |
| jnp | disp8 |
| js | disp8 |
| jns | disp8 |
| loop | disp8 |
| call | addr |
| ret | |

Інші команди:

nop – no operation
hlt – halt operation

di – disable interrupts

ei – enable interrupts

5.4 Програмування перетворювача початкової адреси (ППА)

| № | КОП | адреса в МПП | команда | довжина команди | формати даних |
|----|-----|--------------|-------------|-----------------|------------------------|
| 1 | 01h | 010h | mov R,M | 2 | INT,INTG,REAL,BCD,CHAR |
| 2 | 02h | 020h | mov M,R | 2 | INT,INTG,REAL,BCD,CHAR |
| 3 | 03h | 030h | mov R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 4 | 04h | 040h | mov R,#d16 | 4 | INT,INTG,REAL,BCD,CHAR |
| 5 | 05h | 050h | xchg R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 6 | 06h | 060h | xchg R,M | 2 | INT,INTG,REAL,BCD,CHAR |
| 7 | 07h | 070h | push R | 2 | INT,INTG,REAL,BCD,CHAR |
| 8 | 08h | 080h | pop R | 2 | INT,INTG,REAL,BCD,CHAR |
| 9 | 09h | 090h | add R1,R2 | 2 | INT,INTG |
| 10 | 0Ah | 0A0h | add R,M | 2 | INT,INTG |
| 11 | 0Bh | 0B0h | add M,R | 2 | INT,INTG |
| 12 | 0Ch | 0C0h | add R,#d16 | 4 | INT,INTG |
| 13 | 0Dh | 0D0h | adc R1,R2 | 2 | INT,INTG,CHAR |
| 14 | 0Eh | 0E0h | adc R,M | 2 | INT,INTG,CHAR |
| 15 | 0Fh | 0F0h | adc M,R | 2 | INT,INTG,CHAR |
| 16 | 10h | 100h | adc R,#d16 | 4 | INT,INTG,CHAR |
| 17 | 11h | 110h | addf R1,R2 | 2 | REAL |
| 18 | 12h | 120h | addf R,M | 2 | REAL |
| 19 | 13h | 130h | addf M,R | 2 | REAL |
| 20 | 14h | 140h | addf R,#d16 | 4 | REAL |
| 21 | 15h | 150h | addb R1,R2 | 2 | BCD |
| 22 | 16h | 160h | inc R | 2 | INT,INTG,CHAR |
| 23 | 17h | 170h | sub R1,R2 | 2 | INT,INTG |
| 24 | 18h | 180h | sub R,M | 2 | INT,INTG |
| 25 | 19h | 190h | sub M,R | 2 | INT,INTG |
| 26 | 1Ah | 1A0h | sub R,#d16 | 4 | INT,INTG |
| 27 | 1Bh | 1B0h | subc R1,R2 | 2 | INT,INTG,CHAR |
| 28 | 1Ch | 1C0h | subc R,M | 2 | INT,INTG,CHAR |
| 29 | 1Dh | 1D0h | subc M,R | 2 | INT,INTG,CHAR |
| 30 | 1Eh | 1E0h | subc R,#d16 | 4 | INT,INTG,CHAR |
| 31 | 1Fh | 1F0h | subf R1,R2 | 2 | REAL |
| 32 | 20h | 200h | subf R,M | 2 | REAL |
| 33 | 21h | 210h | subf M,R | 2 | REAL |
| 34 | 22h | 220h | subf R,#d16 | 4 | REAL |
| 35 | 23h | 230h | subb R1,R2 | 2 | BCD |
| 36 | 24h | 240h | dec R | 2 | INT,INTG,CHAR |
| 37 | 25h | 250h | neg R | 2 | INT,INTG,CHAR |
| 38 | 26h | 260h | mul R1,R2 | 2 | INT,INTG |
| 39 | 27h | 270h | mul R,M | 2 | INT,INTG |
| 40 | 28h | 280h | mul M,R | 2 | INT,INTG |
| 41 | 29h | 290h | mul R,#d16 | 4 | INT,INTG |
| 42 | 2Ah | 2A0h | mulf R1,R2 | 2 | REAL |
| 43 | 2Bh | 2B0h | mulf R,M | 2 | REAL |

| | | | | | |
|----|-----|------|-------------|---|------------------------|
| 44 | 2Ch | 2C0h | mulf M,R | 2 | REAL |
| 45 | 2Dh | 2D0h | mulf R,#d16 | 4 | REAL |
| 46 | 2Eh | 2E0h | div R1,R2 | 2 | INT,INTG |
| 47 | 2Fh | 2F0h | div R,M | 2 | INT,INTG |
| 48 | 30h | 300h | div M,R | 2 | INT,INTG |
| 49 | 31h | 310h | div R,#d16 | 4 | INT,INTG |
| 50 | 32h | 320h | divf R1,R2 | 2 | REAL |
| 51 | 33h | 330h | divf R,M | 2 | REAL |
| 52 | 34h | 340h | divf M,R | 2 | REAL |
| 53 | 35h | 350h | divf R,#d16 | 4 | REAL |
| 54 | 36h | 360h | sqr R | 2 | INT,INTG |
| 55 | 37h | 370h | sqr R | 2 | REAL |
| 56 | 38h | 380h | pow R1,R2 | 2 | INT,INTG |
| 57 | 39h | 390h | not R | 2 | INT,INTG |
| 58 | 3Ah | 3A0h | and R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 59 | 3Bh | 3B0h | or R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 60 | 3Ch | 3C0h | xor R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 61 | 3Dh | 3D0h | nxor R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 62 | 3Eh | 3E0h | nand R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 63 | 3Fh | 3F0h | srl R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 64 | 40h | 400h | sll R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 65 | 41h | 410h | sra R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 66 | 42h | 420h | sla R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 67 | 43h | 430h | src R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 68 | 44h | 440h | slc R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 69 | 45h | 450h | cmp R1,R2 | 2 | INT,INTG,REAL,BCD,CHAR |
| 70 | 46h | 460h | cmp R,M | 2 | INT,INTG,REAL,BCD,CHAR |
| 71 | 47h | 470h | cmp R,#d16 | 4 | INT,INTG,REAL,BCD,CHAR |
| 72 | 48h | 480h | jmp addr | 4 | |
| 73 | 49h | 490h | ja disp8 | 2 | |
| 74 | 4Ah | 4A0h | jna disp8 | 2 | |
| 75 | 4Bh | 4B0h | jc disp8 | 2 | |
| 76 | 4Ch | 4C0h | jnc disp8 | 2 | |
| 77 | 4Dh | 4D0h | jz disp8 | 2 | |
| 78 | 4Eh | 4E0h | jnz disp8 | 2 | |
| 79 | 4Fh | 4F0h | jo disp8 | 2 | |
| 80 | 50h | 500h | jno disp8 | 2 | |
| 81 | 51h | 510h | jp disp8 | 2 | |
| 82 | 52h | 520h | jnp disp8 | 2 | |
| 83 | 53h | 530h | js disp8 | 2 | |
| 84 | 54h | 540h | jns disp8 | 2 | |
| 85 | 55h | 550h | loop disp8 | 2 | |
| 86 | 56h | 560h | call addr | 4 | |
| 87 | 57h | 570h | ret | 2 | |
| 88 | 58h | 580h | nop | 2 | |
| 89 | 59h | 590h | hlt | 2 | |
| 90 | 5Ah | 5A0h | di | 2 | |
| 91 | 5Bh | 5B0h | ei | 2 | |

РОЗДІЛ 6. КОМАНДИ ДЛЯ ДЕТАЛЬНОЇ РОЗРОБКИ

Ділення цілих двійкових чисел зі знаком (співмножники та результат - у доповняльному коді), - цілочисельне ділення: ділене, дільник, частка (зі знаком), остача (без знаку)

аccept r10: 0FFFEh

аccept r11: 0FFF6h

link l1: ct

\ Oper1 R11

\ Oper2 R10

\ Res R15

\ Usage r2,r3,r4,r7

{ and r2, r11, 8000h; }

{ and r3, r10, 8000h; }

{ xor r2, r2, r3; }

{ or r15, r2, z; }

{ load rn, flags; and nil, r11, 8000h; }

{ cjp m_z, lp1; }

{ sub r11, z, r11, nz; }

lp1

{ load rn, flags; and nil, r10, 8000h; }

{ cjp m_z, lp2; }

{ sub r10, z, r10, nz; }

equ x: r11

equ y: r10

lp2

{ load rn, flags; sub nil, x, y, nz; } \ x >= y.

{ cjp not m_n, div; }

\ x < y

{ xor r3, r3; }

{ xor r0,r0;}

{ or r0, x;}

{ jmap res; }

div

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 34 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

```

\ division
{ xor r3, r3; }

dloop
{ load rn, flags; sub x, x, y, nz; }
{ cjp m_n, res ; }
{ add r3, 1; }
{ jmap dloop; }

res

\ results
{ or r0,x;}
{load rn,flags; and nil,x,8000h;}
{cjp m_z, go;}
{add r0,r10;}

go
{load rn,flags; and nil,r15,8000h;}
{cjp m_z,respl;}
{sub r15,z,r3,nz;}

respl
{ or r15, r15, r3; }

end{ }

```

Перетворення двійкового коду в двійково-десятковий (для типу даних char)

```

\ PC R10
\ Addr for stos R11
\ Addr for lods R12
\ Source for stos and getter for lods R13
\ Addr for getter R14
\ Addr for source R15

link l1:ct
link l2:rdm
link ewh:16

accept Rdm_Delay:0
accept r15:30h \ source

```

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 35 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

```

dw 30h:13803 \ oper
accept r14:32h \ getter
accept r12:20h \ source lods
accept r11:28h \ getter stos
dw 20h:5,6,7
dw 0:10h \ counter
dw 10h:0,2,0,2,0,2,1 \ steps to lods, stos, conbd
lnext
{ xor nil,r0,r0;oey;ewh;} \ read counter
{ xor nil,r0,r0;oey;ewl;}
m1 { cjp rdm,m1;R;or r0,bus_D,z;}
{ or r10,r0,z;}
{ xor nil,r0,r0;oey;ewh;} \ read command
{ or nil,r0,z;oey;ewl;}
m2 { cjp rdm,m2;R;or r0,bus_D,z;}
{ or nil,r0,z;oey; cjp zo,fp1;} \ if fprog1 jump
{ sub r0,r0,0,z;} \ if fprog2 jump
{ cjp zo,fp2;}
{ sub r0,r0,0,z;} \ if fprog3 jump
{ cjp zo,fp3;}
rets
{ xor nil,r0,r0;oey;ewh;} \ Записуємо новий Program Counter
{ xor nil,r0,r0;oey;ewl;}
m3 { cjp rdm,m3;oey;W;or r10,r10,z;}
{ cjp nz,lnext;}
fp1 \ lods
{ xor nil,r0,r0;oey;ewh;} \ read oper
{ or nil,r12,z;oey;ewl;}
lds1 { cjp rdm,lds1;R;or r13,bus_D,z;}
{ add r12,r12,z,nz;}
{ add r10,z,nz;}
{ cjp nz,rets;}
fp2 \ conbd
{ xor nil,r0,r0;oey;ewh;} \ read oper
{ or nil,r15,z;oey;ewl;}

```

```

bd1 { cjp rdm,bd1;R;or rq,bus_D,z;}
{ and nil,rq,8000h;load rm,flags;}
{ cjp rm_z,sd;}
{ or r3,r3,00f0h;}
{ xor rq,rq,0ffffh;}
{ add rq,rq,1;}
sd
{ xor r4,r4;}
d1
{ add r5,rq,0,z;}
{ sub rq,rq,10000,nz;load rm,flags;}
{ cjp rm_n,d1done;}
{ cjp nz,d1;add r4,r4,nz;}
d1done
{ add rq,r5,z;}
{ or r3,r3,r4;}
{ xor r4,r4;}
{ xor r0,r0;}
{ or r0,r0,1000h;}
d2
{ add r5,rq,0;}
{ sub rq,rq,1000,nz;load rm,flags;}
{ cjp rm_n,d2done;}
{ cjp nz,d2;add r4,r4,r0,z;}
d2done
{ add rq,r5,z;}
{ xor r0,r0;}
{ or r0,r0,0100h;}
d3
{ add r5,rq,0,z;}
{ sub rq,rq,100,nz;load rm,flags;}
{ cjp rm_n,d3done;}
{ cjp nz,d3;add r4,r4,r0,z;}
d3done
{ add rq,r5,z,z;}

```

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| | | | | | | 37 |
| Зм | Арк. | № докум. | Підпис | Дата | | |

```

{ xor r0,r0;}
{ or r0,r0,0010h;}
d4
{ add r5,rq,0;}
{ sub rq,rq,10,nz;load rm,flags;}
{ cjp rm_n,d4done;}
{ cjp nz,d4;add r4,r4,r0,z;}
d4done
{ add rq,r5,z;}
d5
\{ add r5,rq,0;}
{ sub rq,rq,1,nz;load rm,flags;}
{ cjp rm_n,d5done;}
{ cjp nz,d5;add r4,r4,z,nz;}
d5done
{ or r2,r2,r4;}
{ xor nil,r0,r0;oey;ewh;} \ write r2 and r3 to getter
{ or nil,r14,r14;oey;ewl;}
bd2{ cjp rdm,bd2;oey;W;or r2,r2,z;}
{ add r5,r14,z,nz;}
{ xor nil,r0,r0;oey;ewh;}
{ or nil,r5,z;oey;ewl;}
bd3{ cjp rdm,bd3;oey;W;or r3,r3,z;}
{ add r10,r10,z,nz;} \ inc PC
{ cjp nz,end;}
fp3 \stos
{ xor nil,r0,r0;oey;ewh;}
{ or nil,r11,z;oey;ewl;}
sts1{ cjp rdm,sts1;oey;W;or r13,r13,z;}
{ add r11,r11,z,nz;}
{ add r10,z,nz;}
{ cjp nz,rets;}
end{ }

```

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Проектирование цифровых систем на комплектах микропрограммируемых БИС./Под ред. В.Г.Колесникова.-М.:Радио и связь,1984.-240 с,
2. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справ. В 2-х томах./Под ред. В.А.Шахнова.-М.:Радио и связь,1988.
3. Комплект БИС К1804 в процессорах и контроллерах./Под ред. В.Б.Смолова.-М.;Радио и связь,1990.-255 с.
4. Б.А.Калабеков. Микропроцессоры и их применение в системах передачи и обработки сигналов.-М.:Радио и связь, 1988.-368 с.
5. В.И.Корнейчук, В.П.Тарасенко. Вычислительные устройства на микросхемах. Справ. К.Техніка, 1988.-342с.

| | | | | | | |
|----|------|----------|--------|------|--------------------|-----|
| | | | | | ІАЛЦ.467400.003 ПЗ | Арк |
| Зм | Арк. | № докум. | Підпис | Дата | | 39 |