

ВСТУП

Мікропроцесор це електричний пристрій обчислювальної мікроелектронної техніки, який має конструкційну завершеність та програмну налагодженість за рахунок впроваджених складних програмно керованих обчислювальних пристроїв. В мікропроцесорі, в першу чергу, зосереджені пристрої обчислення і обладнання керування процесом програмного обчислення.

Апаратне забезпечення мікросистем є вираженим продуктом у вигляді схем різного рівня. Тому більшість питань читачеві зрозумілі при аналізі складу, будови і роботи схем. Інші обставини виникають при розгляді програмного забезпечення мікропроцесорних систем – продукту, більшу частину якого читач не бачить. Це є невидима оболонка навколо усієї мікропроцесорної апаратури, яка в свою чергу має шаровидний рівневий характер.

На нижньому рівня знаходиться система команд головного мікропроцесора, яка поділена на групи і підгрупи. Кожна команда групи є інструкція процесору для усіх обчислювальних дій. Команди, як і дані та адреси, закодовані в двійковій системі числення. Байтність команд визначається способом адресування та місцем розташування операндів і результату, типом групи і виконуваною функцією. Наступний рівень представлення програмного забезпечення є мови програмування асемблерного типу. Складність програмування у мовному асемблерному програмному середовищі полягає в розумінні інформаційних процесів в пристроях мікросистеми під дією машинних команд. Крім розробки детальних структурних схем алгоритму проєктант повинен володіти майстерністю по підготовці програм на етапах компіляції, компоновки та відлагодження.

Черговий шар програмного забезпечення є мови програмування високого рівня. Оператори мови виступають загально прийняті погодження у вигляді записів відповідних слів. Кожен оператор спричинює виконання в мікросистемі складних процедур, стандартних підпрограм та різноманітних функцій. Останній щабель складного програмного створіння для мікропроцесорів є прикладне програмне забезпечення – комплект взаємо пов'язаних програм для рішення конкретних задач. В цей склад можна

					ДП 5.05010201 435 01 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

включити діагностичні і тестові програми, програмні проекти управління об'єктами, програмне забезпечення різноманітними комунікаційними системами та багато інших програм.

Однокришталеві мікропроцесори є високопродуктивними універсальними пристроями з фіксованим програмним забезпеченням. Архітектура будови гарвардсько – неймановського типу програмно керованого принципу розрахована на використання мікропроцесорів цього класу в виробках обчислювальних систем загального призначення. Однокришталеві мікропроцесори є запозиченою назвою від першого мікропроцесора в зв'язку з розміщенням всієї електроніки в одному кристалі напівпровідникового матеріалу. Більшість виробів систем мікропроцесорного використання, включаючи комп'ютерні системи, збудовані на цьому класі мікропроцесорів.

Секційні мікропроцесори складають обширний набір типів мікропроцесорних секцій визначеного мікропроцесорного комплекту. Секції фіксованої розрядності електрично поєднуються між собою з метою досягнення необхідної розрядної сітки ядра центрального процесорного модуля або схеми мікро програмного керування мікросбірки. Велика продуктивність за рахунок потужної швидкодії обчислень, складність в програмуванні та вузька їх направленість по використанню здебільш дозволили використання цього класу мікропроцесорів в станках числового програмного керування або в спеціалізованих системах управління.

В дипломному проекті висвітлені питання розробки програмно апаратного забезпечення мікропроцесорної системи тестування інтерфейсу пам'яті з головним мікропроцесором. На перший погляд застарілий процесор не створює актуальності в рішенні поставленої задачі, але оригінальність висвітлення розділів проекту, використання математичного апарату логіки при будові програмного та апаратного забезпечення системи створили можливість вільного переходу на більш актуальні процесори. Використання типового алгоритму тестування фізичної пам'яті дозволило побачити помилки в роботі модуля інтерфейсу пам'яті з головним процесором. Крім того, відлагоджена тестова програма в комп'ютері емулює помилку, яка потім цією програмою визначається.

					ДП 5.05010201 435 01 ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

1 ОСНОВНА ТЕХНІЧНА ЧАСТИНА

Структура типової мікропроцесорної системи передбачає розширення її можливостей шляхом нарощення допоміжними модулями, маючими своє призначення. Мікропроцесорна система характеризується **3М (МММ)**: магістральність, модульність, мікропрограмованість.

Магістральність виражається присутністю в системі трьох шин (адреси, даних, керування) – ліній передачі електричних сигналів, які є носіями інформації. Шина даних має два напрямки передачі байтів. Центральний процесор через шину даних отримує з програмної пам'яті команди або дані від портів вводу чи пам'яті даних. Від процесора в пам'ять даних або в порти виводу байти передаються шиною для запису. Напрямок передачі і стан високого імпедансу визначаються автоматично за рахунок вмонтованого в процесор пристрою керування. Шина адреси одно направлена, має шістнадцять ліній з третім станом. Призначена для передачі від процесора зовнішнім пристроям адреси комірок пам'яті чи то портів вводу-виводу, до яких звертається процесор на циклах читання або запису байтів. І, нарешті, шина керування призначена для передачі від процесора адресуємим об'єктам командних сигналів читання або запису фізичної пам'яті чи портів вводу-виводу. Мінімальне число ліній шини складає п'ять – чотири сигнали читання, запису пам'яті та портів вводу-виводу і один читання контролера переривань.

Модульність мікросистеми полягає в її складі у вигляді окремих модулів, до яких віднесені модулі центрального процесора, пам'яті, вводу-виводу, різноманітних адаптерів чи то контролерних пристроїв, маючих інтерфейс з системною магістраллю. Системну магістраль, у свою чергу, складають шини адреси, даних, керування. Кожен із модулів призначено і орієнтовано на виконання конкретних функцій, здебільш за все дуже складних.

Мікропрограмованість модулів визначається їх програмною налагодженістю за рахунок початкової ініціалізації та програмно керованої роботи в ході обчислень. Модуль ядра центрального процесора сумісно з модулем пам'яті реалізують закладену в систему функцію принципу програмного керування.

					ДП 5.05010201 435 01 ПЗ	Арк.
						7
Змін.	Арк.	№ докум.	Підпис	Дата		

1.1 Склад інтерфейсу адресованих модулів з головним мікропроцесором

Магістрально модульна відкрита структура мікропроцесорної системи передбачує підключення до системної магістралі певну кількість модулів пам'яті та необхідні допоміжні мікропроцесори. Ізольований від фізичної пам'яті просторій введення - виведення допускає підключення до шини головного процесора до 256-ти портів введення та 256-ти портів виведення. Залежно від типу периферійної мікропроцесорної системи, яка має фізичний інтерфейс з магістраллю, витрачається, або іншими словами, завантажується декілька портів. Мікропроцесорні модулі мають типову структуру в плані можливостей будови інтерфейсу з головним процесором. Для вибору мікросхем типових мікропроцесорів, як втім і мікросхем пам'яті, використовуються зовнішні дешифратори у вигляді комбінаційних пристроїв або програмуючих матриць. Несумісність командних сигналів запису порту \overline{IOWC} та читання селектованого порту введення \overline{IORC} , сформованих системним контролером процесорного модуля, обумовлена виконанням команд виведення та введення відповідно.

Структура мікросхем пам'яті, крім виводів керування читанням та записом даних, передбачає присутність лінії на виводі, по якій передається сигнал вибору, сформований адресним селектором. Системна шина керування модуля процесора має сигнали читання \overline{MRDC} та запису \overline{MWTC} пам'яті, які з'єднуються з відповідними входами мікросхеми. Організація комірок пам'яті частіше є словарною розрядністю в один байт. Їмність мікросхеми пам'яті визначається кількістю комірок та зв'язана з числом входів введення адреси. Для базового процесора максимальна ємність визначається в 64 К байти. Певна мікропроцесорна системи має у своєму складі від одиниць до декілька десятків мікросхем оперативної та постійної пам'яті. У цьому адресному просторі розміщується пам'яті даних, програмна пам'яті та стекова пам'яті. При виконанні команд в процесорному модулі функції пам'яті розмежені таким чином, що автоматично головний мікропроцесор

					ДП 5.05010201 435 01 ПЗ	Арк.
						8
Змін.	Арк.	№ докум.	Підпис	Дата		

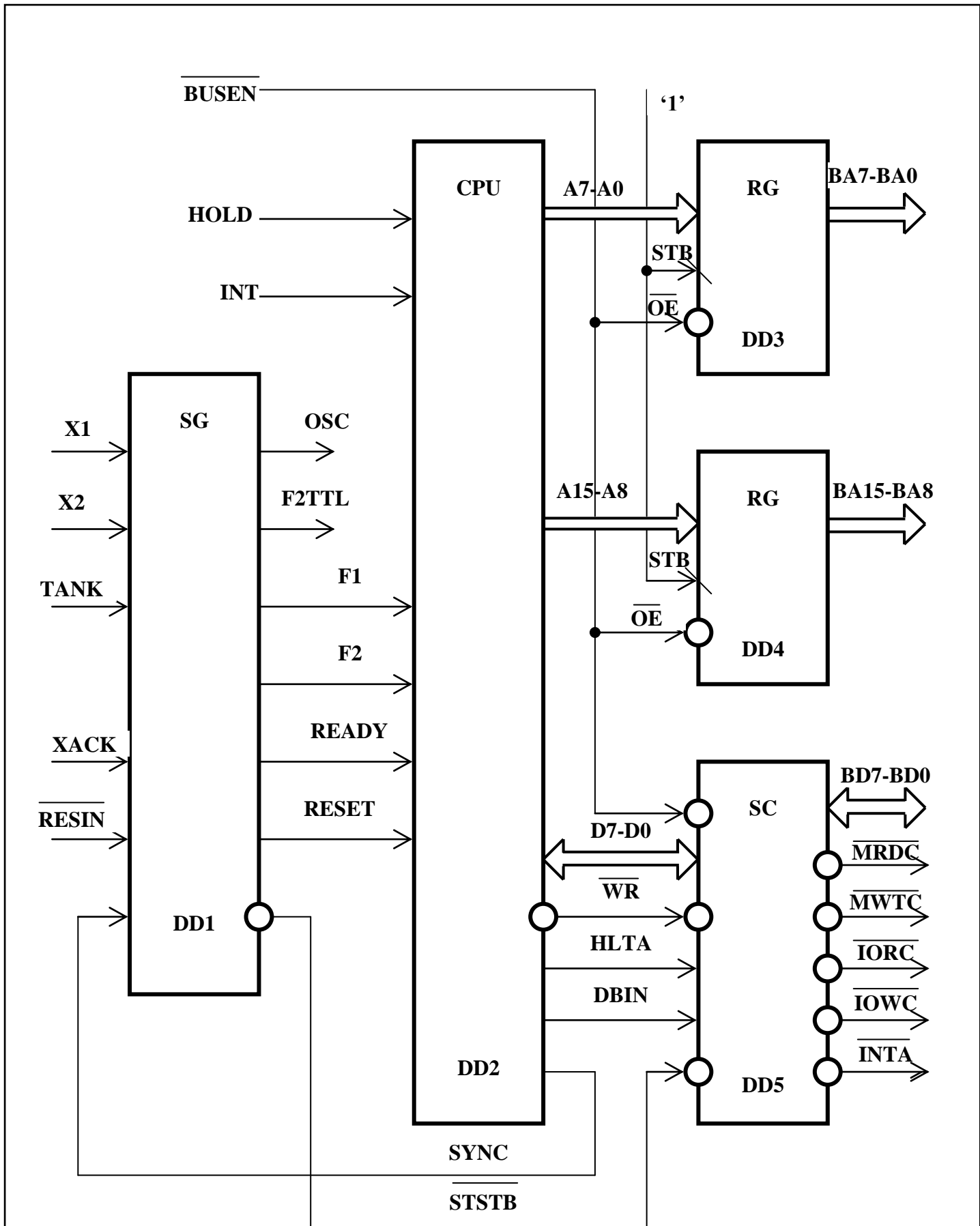


Рисунок 1 – Схема електрична структурна модуля центрального процесора

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

9

звертається за певною адресою виключно до відповідного типу пам'яті. Усе це передбачено закладено в програмно керований принцип обробки команд.

Для будови процесорного модуля розробниками були спроектовані та запроваджені на використання допоміжні мікросхеми системного генератора, системного контролера магістралі даних і керування, буферний регістр. Схема електрична функціональна модуля приведена на малюнку Додаток 2.

Системний генератор **SG (System Generator)** забезпечує роботу головного мікропроцесора шляхом подачі неперериваючихся в часу тактових імпульсів **F1, F2**. Для допоміжних пристроїв мікропроцесорної системи формуються в генераторі також сигнали управління **F2TTL**, частота яких зменшена в два рази за частоту **F2**. Як відмічалося раніше, генератор передає процесору строго в потрібному часовому терміні сигнали готовності **READY** і скиду **RESET**, який використовується другими мікропроцесорами системи. Для запису видане на зовні із мікропроцесора слово стану в системний контролер генератор формує сигнал стробу стану процесора **STSTB#**.

Системний контролер **SC (System Controller)** магістралі забезпечує формування двох магістралей – даних та керування. Зовнішній сигнал зайнятості - дозволу **BUSEN# (BUSy-ENable)** має низький рівень, що забезпечує нормальну роботу контролера. Він формується пристроєм арбітра магістралі. При високому рівні на керуючому виході усі виходи контролера перемикаються в третій стан, що спроваджує призупинену в роботі головного процесора. У випадку блокованого стану процесора сигналом **HLDA** від мікропроцесора магістралі даних і керування також приймають стан високого імпедансу. Напряму передачі байтів через магістраль даних визначається двома сигналами **WR#, DBIN**, які керують також процесом формування магістралі керування. Процес читання чи запису пам'яті або портів синхронізован цими сигналами, як втім і визначається напрям передачі магістралі даних. Контролер має пристрій пам'яті для збереження терміном в машинний цикл слова стану процесора, яке завантажується по сигналу від генератора тактових імпульсів **STSTB#**. В свою чергу робота генератора по формуванню строба синхронізована головним мікропроцесором.

					ДП 5.05010201 435 01 ПЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис	Дата		

1.1.1 Інтерфейс портів введення виведення з модулем центрального процесора мікросистеми

У спрощених мікропроцесорних системах порти використовуються для розміщення програмно доступних регістрів периферійних мікропроцесорів, які мають стандартний інтерфейс з процесорним ядром.. Системна магістраль формується за рахунок використання допоміжних мікросхем. Так магістраль адреси **BA15-BA0** передає через буфер **BA** активну адресу протягом усього машинного циклу. Елементи пам'яті для будови буферу не використовуються тому, що локальна шина адреси головного мікропроцесора ізольована від інших виводів і зберігає протягом усього машинного циклу адресу. Буфер даних **BD** двонаправлений, на відміну від буфера адреси, отримує від головного процесора сигнали управління вибору буферу та керування напрямком передачі даних. Розрядність буферу складає вісім біт **BD7-BD0**. Через буфер процесор отримує команди від програмної пам'яті, дані - від пам'яті даних та від стеку або портів введення. При виведенні в пам'ять даних або порти виводу буфер прозорий від головного процесора в зовнішнє середовище.

Формування командних сигналів читання \overline{IORC} та запису \overline{IOWC} портів введення виведення виконується з урахуванням певних ознак слова стану процесора та відповідних стробів, які процесор залежно від напрямку передачі байту відповідно формує. Логіка формування сигналів управління визначена рівняннями та відповідною схемою

$$IORC = INP \& DBIN;$$

$$\overline{IORC} = \overline{INP \& DBIN} ;$$

$$IOWC = OUT \& WR;$$

$$\overline{IOWC} = \overline{OUT \& WR} = \overline{OUT} \vee \overline{WR}.$$

Дешифратор адреси вибирається у випадку, якщо процесор пише або читає порти. Формування вибору $\overline{CS_{DC}}$ порту виконується клапаном кон'юнкції

$$CS_{DC} = IORC \vee IOWC;$$

$$\overline{CS_{DC}} = \overline{IORC \vee IOWC}.$$

					ДП 5.05010201 435 01 ПЗ	Арк.
						11
Змін.	Арк.	№ докум.	Підпис	Дата		

Ураховуючи вище описаний принцип будови, спрощена схема підключення портів введення виведення до системної магістралі має такий вигляд.

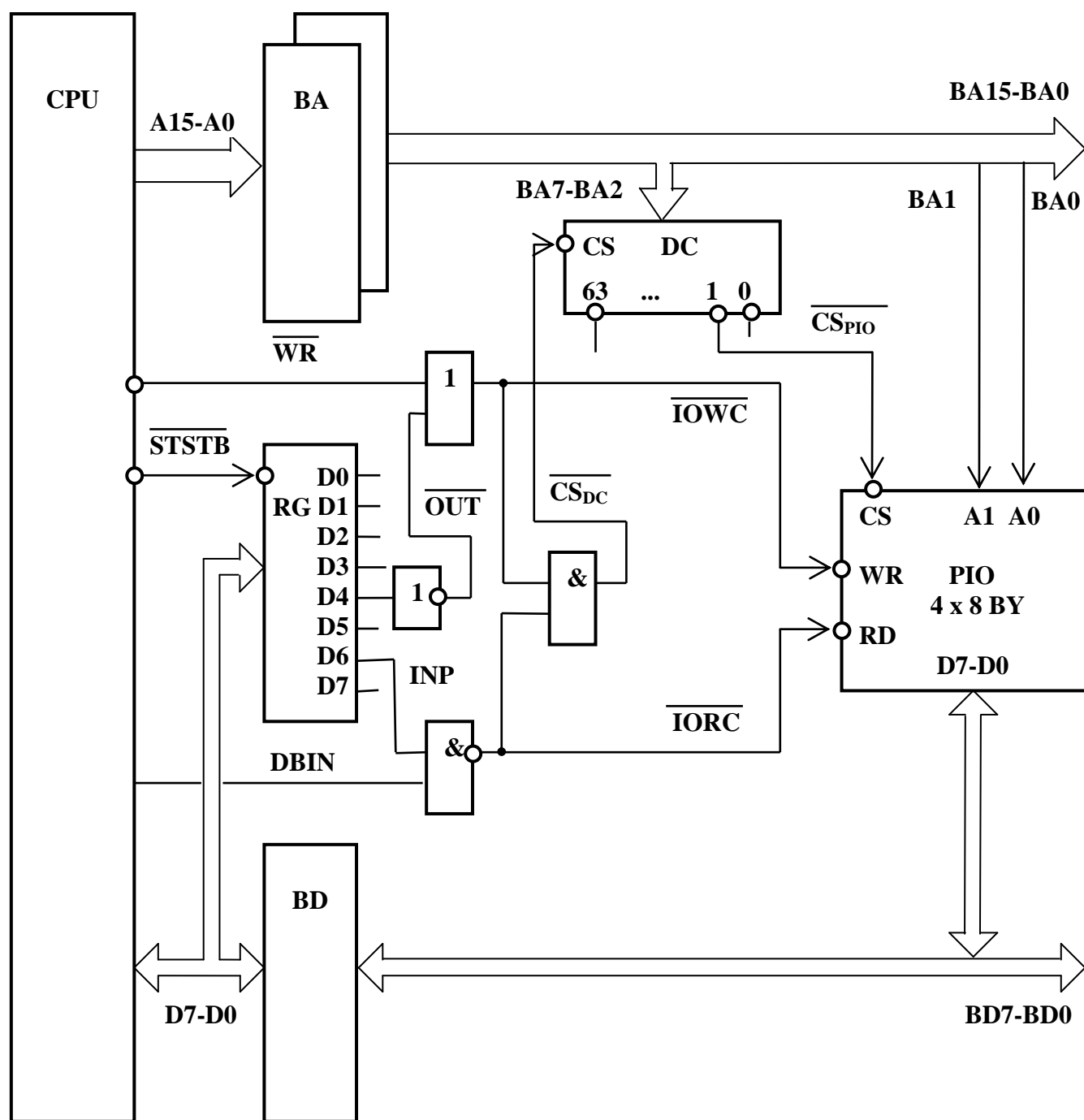


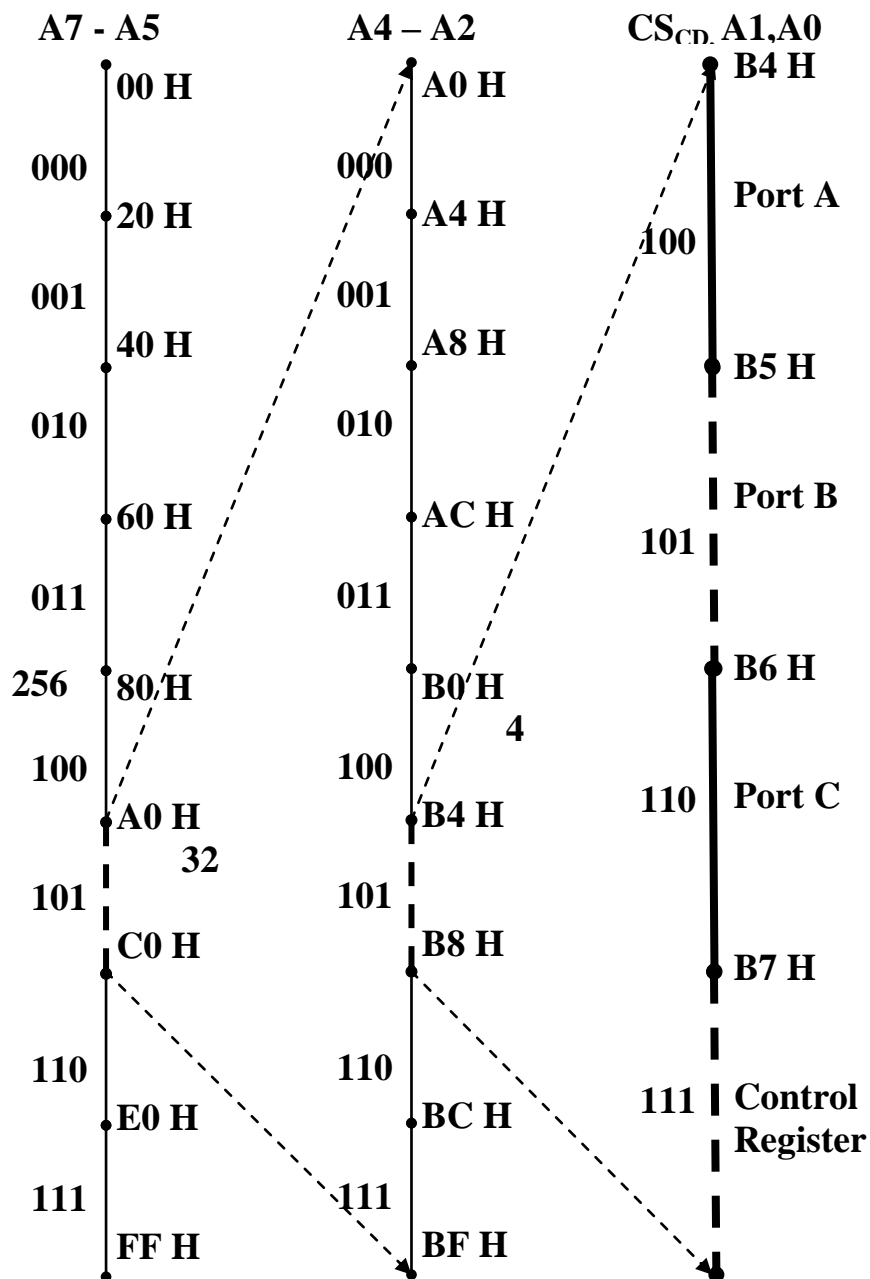
Рисунок 2 - Інтерфейс портів з модулем центрального процесора

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

12



$$\begin{aligned}
 CS_{PI} &= (CS_{CD} \& A7) \& \overline{A6} \& A5 \& (A4 \& \overline{A3} \& A2) = \\
 &= (\overline{CS_{CD}} \vee \overline{A7} \& \overline{A6} \& A5) \& F5; \\
 \overline{CS_{PI}} &= (\overline{CS_{CD}} \vee \overline{A7} \& \overline{A6} \& A5) \& F5.
 \end{aligned}$$

Малюнок 3 – Карта пам'яті розміщення портів в фізичному просторі та логіка керування дешифратором вибору

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

13

На верхньому малюнку показана карта пам'яті розміщення мікросхеми пам'яті за адресою **6800 H** ємністю два кілобайти. За допомогою карти отримані логічні рівняння для функції вибору мікросхеми пам'яті в допущенні використання мікросхем середнього ступеню інтегрованості. Ураховуючи попередні тлумачення відносно формування командних сигналів \overline{IORC} і \overline{IOWC} та отримані рівняння, унизу побудована спрощена схема підключення мікросхеми пам'яті до системної магістралі мікросистеми.

При підключенні до системи адаптера паралельного інтерфейсу ураховані обставини адресування чотирьох портів адаптеру, які адресуються молодшими лініями адреси **BA1, BA0**. Вибір адаптеру через дешифратор адреси виконується старшими розрядами магістралі адреси **BA7 – BA2**. Звернення до портів введення виведення по командам процесора введення виведення виконується з урахуванням того, що адреса активного порту передається з процесора дубльовано – молодша та старша половини магістралі адреси дублюються. Для вибору адаптеру необхідна бути присутня в шині адреси певна інформація про адресу. Це дає можливість активізувати роботу дешифратора. При зверненні до портів активна адреса від процесора дублюється по молодшій та старшій половині магістралі адреси. Дешифратор спрацює лише в умові читання або запису порту, коли активний один із командних сигналів \overline{IORC} , \overline{IOWC} .

Згідно з приведеною картою розміщення адаптера в просторі введення виведення активними портами є чотири **(B4-B7) H**. Допоміжна комбінаційна логіка, включаючи дешифратор адреси вибору адаптера, забезпечує вибір необхідного порту та регістра управління адаптера. Управління логікою виконується зі сторони головного процесора та від регістру збереження слова стану процесора. Процесор передає один із активних стробів читання **DBIN** або запису \overline{WR} . Від регістру логіка отримує активну ознаку слова стану, яка визначає тип текучого машинного циклу. Несумісність стробів на текучому циклі забезпечує одно визначеність по запису даних в порт або по читанню та виключає конфліктність на шині даних. Усі ці дії автоматизовані і передбачені структурно – архітектурно при проектуванні головного процесора.

					ДП 5.05010201 435 01 ПЗ	Арк.
						14
Змін.	Арк.	№ докум.	Підпис	Дата		

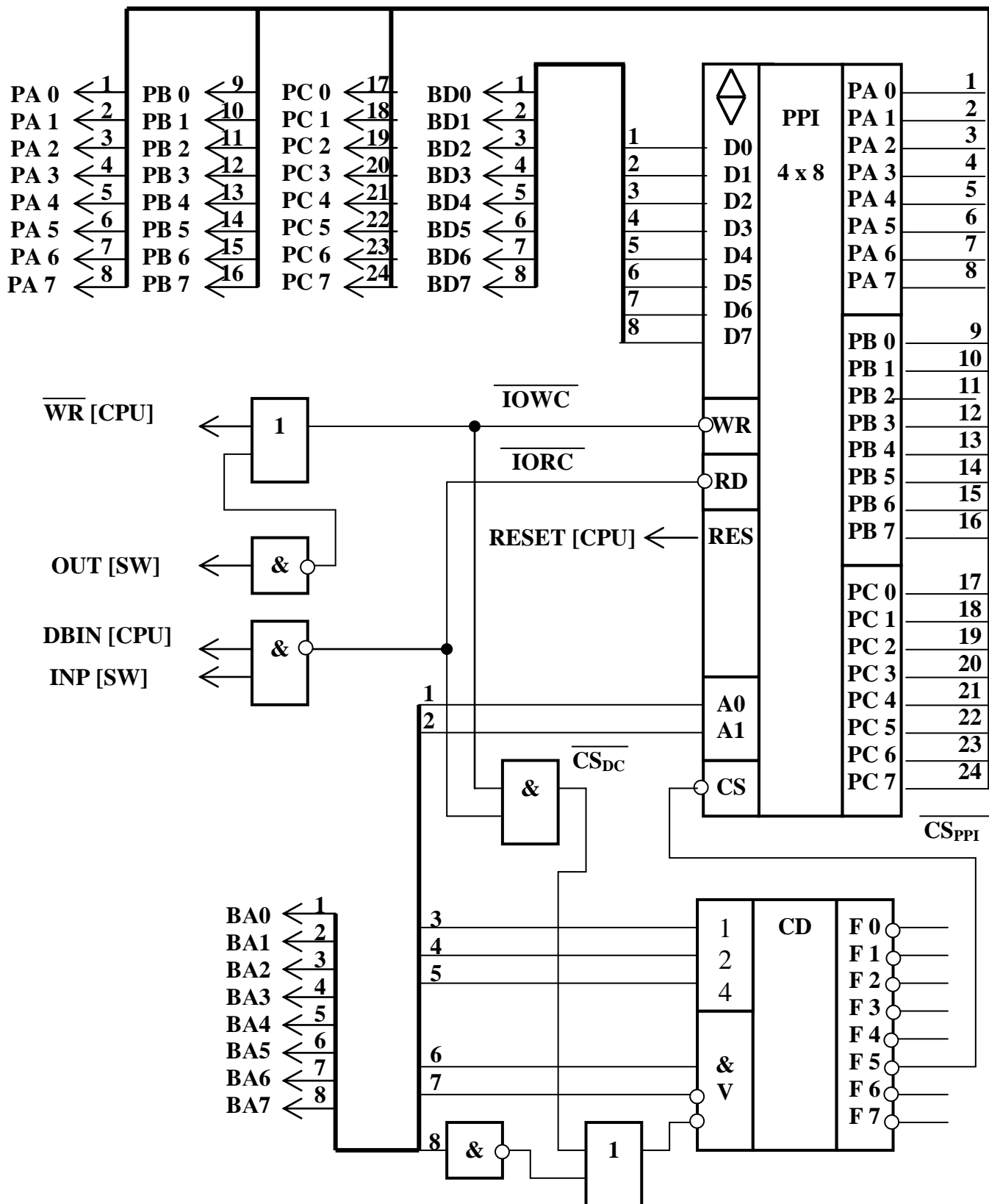


Рисунок 4 – Фрагмент схеми інтерфейсу портів з системною магістраллю мікропроцесорної системи

1.1.2 Інтерфейс портів контролера переривань з центральним процесорним модулем

Контролер переривань є важливою мікропроцесорною схемою в будові та використанні типових мікросистем. Він є колектором – збирачем усіх запитів на переривання головного процесора з метою обслуговування кожного. Запитом на переривання є прапор готовності зовнішнього пристрою, який має інтерфейс з системною магістраллю процесорного модуля. Оскільки пристроїв може бути декілька з асинхронним непередбаченим запитом по відношенню до роботи головного процесора, то виникає проблема їх послідовності обслуговування. Програмованість контролера переривань передбачає запис в регістри контролера векторів переривань та послідовність їх обслуговування. Вхідні запити зберігаються в контролері, а підключений механізм пріоритетного обслуговування дозволяє формувати в процесорне ядро відповідний запит на переривання.

З програмної точки зору контролер переривань на перший погляд має усього два програмно доступні регістри тому, що вони адресуються однією лінією адреси **BA0**. Але це не так, оскільки при програмуванні контролера повинна витримуватися певна послідовність при завантаженні слів ініціалізації та слів управління. Тут спрацьовує допоміжна логіка вибору регістрів, яка для користувача прихована.

Контролер, як втім і інша периферійна мікропроцесорна схема, повинен бути адресованим з боку певного дешифратора, який декодує старші лінії магістралі адреси **BA7-BA1**. Формування командних сигналів читання \overline{IORC} та запису \overline{IOWC} портів контролера виконується з урахуванням певних ознак слова стану процесора та відповідних стробів, які процесор залежно від напрямку передачі байту відповідно формує. Командний сигнал читання виклику вектора переривання \overline{INTA} формується за певною одно іменною ознакою слова стану процесора та стробовим сигналом від процесора **DBIN [CPU]**. Логіка формування сигналів управління визначена логічними рівняннями та відповідною схемою.

					ДП 5.05010201 435 01 ПЗ	Арк.
						16
Змін.	Арк.	№ докум.	Підпис	Дата		

1.2 Структура головного мікропроцесора мікросистеми

Обладнання мікропроцесора можна поділити на операційні частини, головними із яких є арифметико логічний пристрій, файл регістрів загального призначення, пристрій керування, буфери адреси і даних. Самі назви говорять за себе, але є деякі приховані структурно - архітектурні особливості.

Арифметичний пристрій (*ALU – Arithmetic and Logic Unit*) має архітектуру обчислень акумуляторного типу. Більшість арифметико-логічних команд адресують другий операнд, передбачаючи, що перший знаходиться в головному накопичувачі – акумуляторі (*A -Accumulator*). Результат обчислень зберігається в акумуляторі, замінюючи перший операнд. Крім зафіксованого результату, в регістрі прапорів (*F – Flags*) фіксуються п'ять прапорів, які є ознакою отриманого в акумуляторі результату. Передбачена можливість програмного тестування істинного та оберненого значень чотирьох із них. Це дає змогу будувати циклічні та петлюючі алгоритми, що є дуже потужним зрушенням в програмуванні.

Файл регістрів загального призначення насичено не дуже великим їх числом, але наявність такого сховища дуже сильно розв'язує руки програмісту. Регістри файлу тимчасово можуть по вимогам програми зберігати байти даних, регістрові пари дають можливість зберігати та програмно передбачено адресувати байти даних в пам'яті мікропроцесорної системи. Два важливих програмно недоступних прихованих регістрів *W, Z* складають пару *WZ*, яка виконує адресну функцію. Зв'язок регістрів з внутрішньою магістраллю мікропроцесора підтримано через мультиплексор (*MX - Multiplexor*).

Показник стеку (*SP – Stack Pointer*) відіграє роль адресного регістру та показує на вершину стеку – комірку пам'яті сегменту стека, куди при останньому зверненні процесора було записано слово із двох байтів. Цей регістр ніяким чином не взаємодіє з АЛП тому, що зменшення або збільшення на одиницю його стану визначається схемою інкременту-декременту. Класична будова стеку, а вона тут впроваджена, передбачає перед записом байту в стек стан *SP* зменшується на одиницю. Після читання байту зі стеку стан *SP* збільшується на одиницю.

					ДП 5.05010201 435 01 ПЗ	Арк.
						18
Змін.	Арк.	№ докум.	Підпис	Дата		

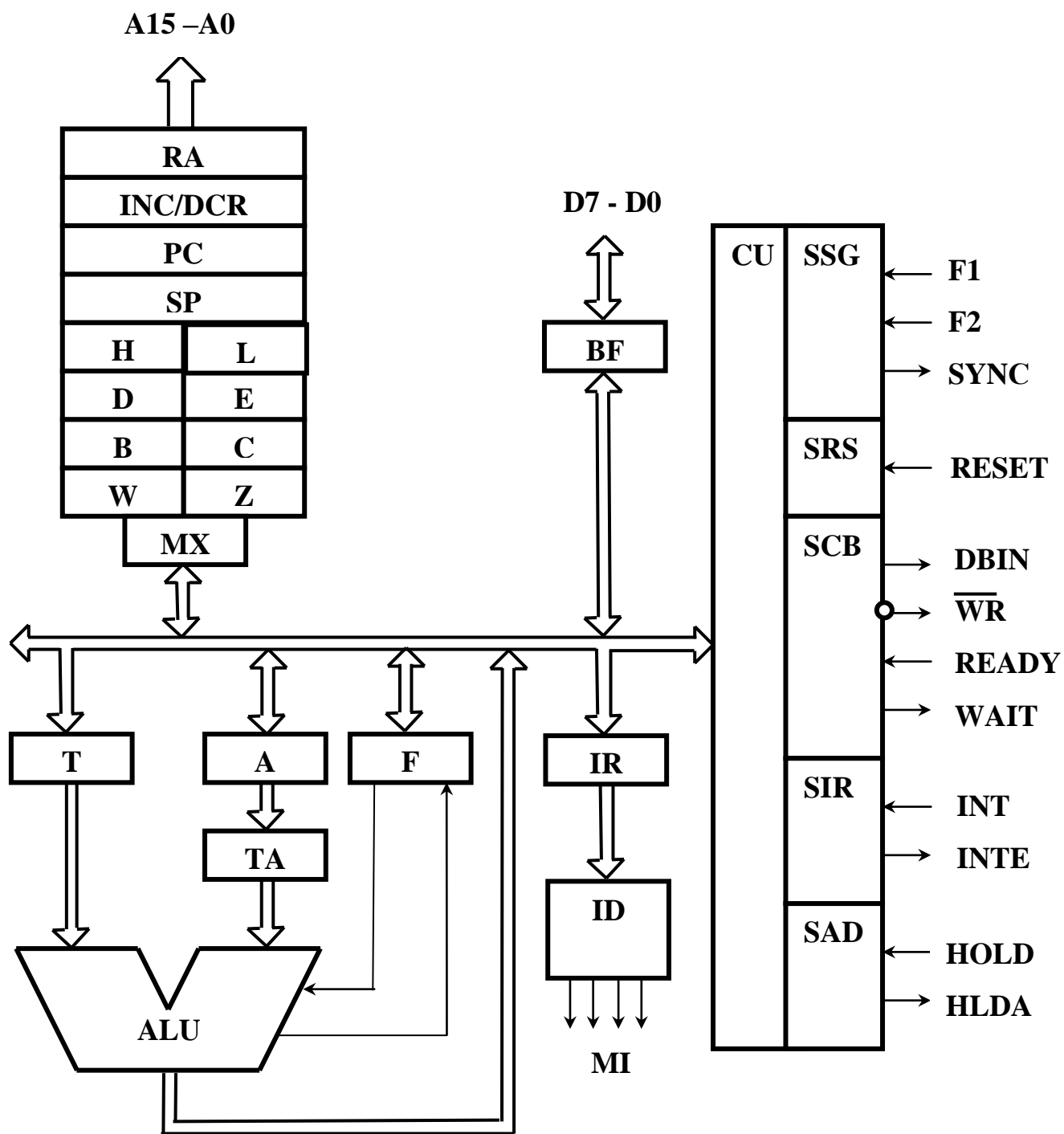


Рисунок 6 - Структурна схема мікропроцесора i8080

Програмний лічильник (*PC – Program Counter*) не поділено на байти, але при передачі його в стек перед перериваннями чи викликах процедур умовно є ділення на молодшу і старшу частини. При зверненні до стеку маніпуляції виконуються над байтами. Призначення лічильника команд загально відоме – це формування адреси програмної пам'яті наступного байту командного потоку за рахунок збільшення на

					ДП 5.05010201 435 01 ПЗ	Арк.
						19
Змін.	Арк.	№ докум.	Підпис	Дата		

одиницю в схемі **INC/DCR**. Команди галуження можуть змінювати стан лічильника шляхом загрузки адреси питлювання.

Регістр адреси (**RA – Register Address**) є приймачем адреси від програмного лічильника, показника стеку, будь-якої реєстрової пари. Для зовнішніх пристроїв він є джерелом адреси комірки пам'яті чи порту вводу-виводу. Виходи реєстру мають третій стан, що дає змогу відключати локальну магістраль від зовнішніх пристроїв.

Пристрій управління (**CU - Control Unit**) має складну структуру внутрішньої схемотехнічної будови. Пов'язано це з тим, що він сприймає вхідні зовнішні сигнали, формує вихідні сигнали керування і внутрішні сигнали мікрооперацій (**MI – MicroInstruction**). Обладнання пристрою управління складається з надсистем, які забезпечують відповідні режими роботи (режим програмної обробки команд мікропроцесора, режим переривання і блокованого стану при прямому доступі процесора до пам'яті, стан скиду в початкову роботу процесора). Складність пристрою обумовлена контролем процесом обміну даними між ядром мікропроцесорного обчислювача і пам'яттю та портами вводу-виводу, програмно керованим обчисленням команд та інформації про адреси програмно доступних внутрішніх і зовнішніх елементів. Тип пристрою управління має структуру часової організації програмно керованого управління обчисленнями синхро-асинхронної дії. Синхронізм полягає в часу існування типів машинного циклу, асинхронізм – в часу протікання командних циклів.

Зв'язок мікропроцесора з зовнішніми пристроями по передачі інформації за відповідним напрямком забезпечується через буфер (**BF –Buffer**), маючий керований третій стан. Напрямок передачі завжди визначає мікропроцесор з метою контекстного та безконфліктного обміну байтами з зовнішнім середовищем. Для забезпечення нормальної роботи головного мікропроцесора передбачена передача зовнішніх сигналів керування по дванадцятих лініях. Крім того, використовуються ще двадцять чотири лінії, які складають локальні магістралі адреси і даних.

					ДП 5.05010201 435 01 ПЗ	Арк.
						20
Змін.	Арк.	№ докум.	Підпис	Дата		

Тактові сигнали ***F1***, ***F2*** формують зовнішній генератор, функції якого полягають в забезпеченні правильної роботи по синхронізації процесів перемикання внутрішніх операційних пристроїв мікропроцесора. Вимогою часових параметрів сигналів тактування є їх дія, як це показано на малюнку.

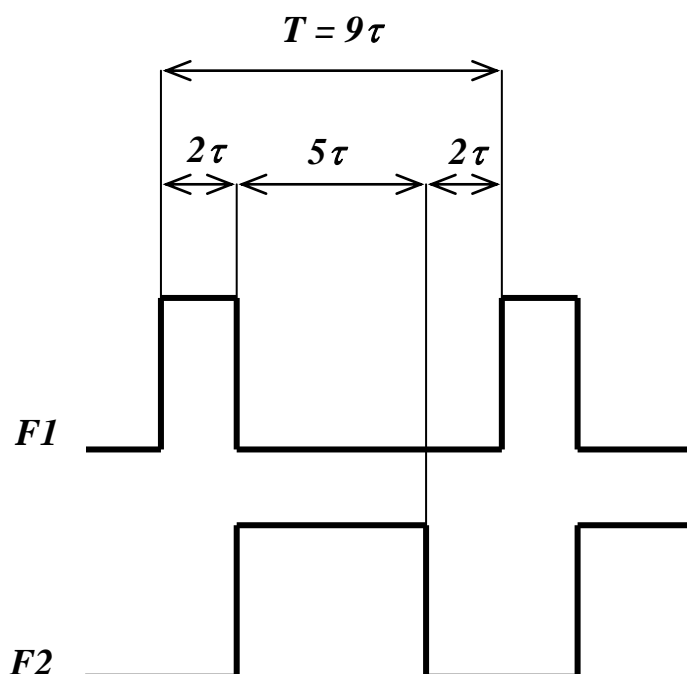


Рисунок 7 – Діаграма формування тактових імпульсів ***F1***, ***F2***

Період сигналів тактування ***F1*** прийнято називати мікротактом або станом мікропроцесора. Період дорівнює $T = 9\tau$ (τ – період опорних сигналів осцилятора системного генератора). Тривалість сигналів відповідно дорівнюють 2τ і 5τ , а між ними присутня щілина тривалістю 2τ . Усе це обумовлено схемотехнікою будови внутрішніх пристроїв мікропроцесора. На кожному такті виконуються складні перемикаючі процеси, в зв'язу з чим мікропроцесор приймає новий свій стан.

Вихідна лінія сигналу ***SYNC*** (***SYNChronization***) завжди на першому такті машинного циклу передає зовнішнім пристроям мікросистеми сигнал синхронізації. Тому усі перемикаючі процеси в пристроях синхронізовані. Сигнал синхронізації стробован в мікропроцесорі фронтом тактових імпульсів ***F2***.

					ДП 5.05010201 435 01 ПЗ	Арк.
						21
Змін.	Арк.	№ докум.	Підпис	Дата		

Призначення сигналу **RESET** відоме - для переведу процесора в початковий стан після подачі напруги живлення або після скиду від спеціальної кнопки. Сигнал **RESET** формується в системному генераторі після вище названих подій і синхронізований зрізом тактових імпульсів **F2**. Тривалість сигналу повинна перевищувати самий довгий в часу машинний цикл, що дорівнює п'ять тактів головної синхронізації. Після дії сигналу **RESET** програмний лічильник перемикається в ноль, забороняються зовнішні переривання, мікропроцесор переходить в стан **T1** машинного циклу читання першого байту команди з програмної пам'яті. З цього починається робота мікропроцесора.

Вихідні сигнали **DBIN (Data Bus Input)**, **WR# (Write)** забезпечують читання, запис байтів відповідно в фізичній пам'яті або портах вводу-виводу. На текучому магистральному циклі сигнали несумісні, так як обидва одночасно активними бути не можуть. Керуючий сигнал прийому **DBIN** активний на такті **T2** відповідного типу машинного циклу і стробований фронтом **F2**. Сигнал запису **WR#** стробований фронтом **F1** і активний на третьому такті низьким рівнем відповідного типу машинного циклу. Обидва керуючі сигнали приймають участь в формуванні в зовнішньому контролері магистралі керування мікропроцесорної системи.

Сигнал готовності **READY** на відповідному вході мікропроцесора формується в зовнішньому генераторі, який, у свою чергу, отримує від пристроїв пам'яті чи то портів готовність до обміну (читання або запис байту) і забезпечує синхронізацію видачі готовності в мікропроцесор. Готовність **READY** в генераторі стробована фронтом **F2** і передається в процесор на такті **T2** текучого командного циклу. Зрізом **F2** на другому такті завжди процесором аналізується сигнал готовності. Якщо **READY = 1**, то процесор безперешкодно перемикається на стан **T3** для завершення передачі байту. У випадку неготовності **READY = 0** процесор перемикається в стан очікування готовності **TW** зовнішніх повільнодіючих пристроїв до обміну байтами зі швидкодіючим процесором. Між тактами **T2** і **T3** вкладаються такти очікування **TW** до появи сигналу **READY = 1**.

Вихідний сигнал очікування **WAIT** інформує зовнішні пристрої про стан процесора очікування готовності зовнішніх пристроїв до обміну байтами. В стані процесора **TW** сигнал **WAIT = 1**. Сигнал очікування стробований фронтом **F1**. При деяких других станах мікропроцесора сигнал **WAIT** також активізується.

					ДП 5.05010201 435 01 ПЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис	Дата		

Впроваджена система маскованих багаторівневих переривань мікропроцесора пов'язана з вихідним програмно-апаратно керованим сигналом дозволу зовнішнього переривання **INTE (INTerrupt Enable)**. Програмно сигнал можна модифікувати. Якщо **INTE = 0**, то запити зовнішніх переривань мікропроцесором ігноруються. У випадку переривання мікропроцесора сигнал **INTE** автоматично скидається, блокуючи вкладеність наступного переривання в текуче. Для дозволу вкладеності переривань процедура текучого переривання повинна мати команду установки в одиницю сигналу **INTE**.

Запит на переривання **INT (INTerrupt request)** формується в контролері переривань і поступає в процесор непередбачено. Умовою переривання мікропроцесора є логічне рівняння **INT & INTE = 1**. При виконанні умови переривання мікропроцесор закінчує текучий командний цикл (закінчує виконання команди) і складним способом від контролера переривань отримує вектор переривань, а сформовану адресу в **PC** наступної команди стандартним способом завантажує в стек. Вектор (адреса перекриваючої процедури) завантажується в програмний лічильник.

Сигнал блокування процесора **HOLD** поступає від контролера прямого доступу до пам'яті. Стан процесора блокування призупиняє читання команд з пам'яті та переводом локальної магістралі в третій стан імпедансу. Після зняття сигналу **HOLD** процесор поновлює свою роботу згідно з програмою.

Блокований стан процесора відмічається видачею сигналу підтвердження **HLDA (HoLD Acknowledge)**, який поступає в контролер прямого доступу. При прямому доступі до пам'яті головний процесор блокований і не виконує ніяких обчислювальних дій за виключенням внутрішніх пересилок або обчислень. Системна магістраль при цьому знаходиться у третьому стані. Контролер прямого доступу генерує магістральні цикли передачі байтів між пам'яттю та периферійним пристроєм. Така передача по певному напрямку виконується усього за чотири такти головної синхронізації, що збільшує швидкість каналу зв'язку.

1.2.1 Будова і склад арифметико логічного пристрою мікропроцесора

Відомо, що чотири основні арифметичні дії в процесорі виконуються за рахунок базової мікрооперації складання, реалізація якої можлива у суматорі. Базова логіка

					ДП 5.05010201 435 01 ПЗ	Арк.
						23
Змін.	Арк.	№ докум.	Підпис	Дата		

(кон'юнкція, диз'юнкція, заперечення) дозволяє реалізовувати більш складну логічну функцію, що відображено групою системи команд мікропроцесора. В зв'язку з цим суматор, як головний операційний пристрій, повинен мати допоміжну логічну схему виконання трьох логічних базисів. Одна арифметична і три логічних мікрооперацій складають функційно повний базис арифметико-логічних команд мікропроцесора. Треба також мати на увазі, що заперечення є унарною мікрооперацією, що для суматора арифметико логічного пристрою процесора з його розвинутою логічною схемою є визначною складністю. Тому ця дія замінена бінарною операцією логічної нерівності, першим операндом якої є одиниця, а другим – аргумент. Сума по модулю два одиниці з аргументом дає значення заперечення.

Роботу однорозрядного суматора на три входи і два виходи можна представити двома мінімізуючими картами Карно. Це дає змогу спростити логічні функції формування суми S_i і перенесення у старший розряд C_i , залежно від аргументів – двох операндів X_i, Y_i і перенесення з молодшого розряду C_{i-1} .



Рисунок 8 - Карти Карно однорозрядного суматора на три входи

Після склеювань у клітинах карти для перенесення та відповідних математичних перетворень визначаються досить цікаві логічні рівняння вираження функцій суми і перенесення у старший розряд трьома бінарними базовими логічними функціями.

$$\begin{aligned}
S_i &= \overline{X_i} \overline{Y_i} C_{(i-1)} \vee \overline{X_i} Y_i \overline{C_{(i-1)}} \vee X_i \overline{Y_i} \overline{C_{(i-1)}} \vee X_i Y_i C_{(i-1)} = \\
&= (\overline{X_i} \overline{Y_i} \vee X_i Y_i) C_{(i-1)} \vee (\overline{X_i} Y_i \vee X_i \overline{Y_i}) \overline{C_{(i-1)}} = \\
&= \overline{M_i} C_{(i-1)} \vee M_i \overline{C_{(i-1)}} = M_i \oplus C_{(i-1)};
\end{aligned}$$

$$C_i = X_i Y_i \vee X_i C_{(i-1)} \vee Y_i C_{(i-1)} = X_i Y_i \vee (X_i \vee Y_i) C_{(i-1)} = K_i \vee D_i C_{(i-1)};$$

Функція суми $S_i = M_i \oplus C_{(i-1)};$

Функція нерівнозначності $M_i = X_i \oplus Y_i;$

Функція кон'юнкції $K_i = X_i Y_i;$

Функція диз'юнкції $D_i = X_i \vee Y_i;$

Функція перенесення $C_i = K_i \vee D_i C_{(i-1)}.$

Логічний базис кон'юнкції **Ki**, диз'юнкції **Di**, суми по модулю два **Mi** дозволяє будувати однорозрядний суматор, формуючи суму **Si**. Всі ці чотири функції, складаючі обчислювальний базис мікропроцесора, можна поєднати з використанням мультиплексора. Опис роботи мультиплексора на чотири інформаційні і два керуючі **A**, **B** входи та один функційний вихід **Fi** має відповідне рівняння.

$$F_i = (\overline{B} \cdot \overline{A}) \& S_i \vee (\overline{B} \cdot A) \& M_i \vee (B \cdot \overline{A}) \& D_i \vee (B \cdot A) \& K_i.$$

Входи керування, у свою чергу, формуються шифратором, який кодує арифметико-логічні команди мікропроцесора. До цих команд відносяться три логічних (кон'юнкція **ANA**, диз'юнкція **ORA**, нерівнозначність **XRA**) і одна арифметична, визначаюча близько типові інструкції мікропроцесора складання без урахування **ADD** та з врахуванням **ADC** перенесення віднімання без урахування **SUB** та з врахуванням **SBB** позики.

					ДП 5.05010201 435 01 ПЗ	Арк.
						25
Змін.	Арк.	№ докум.	Підпис	Дата		

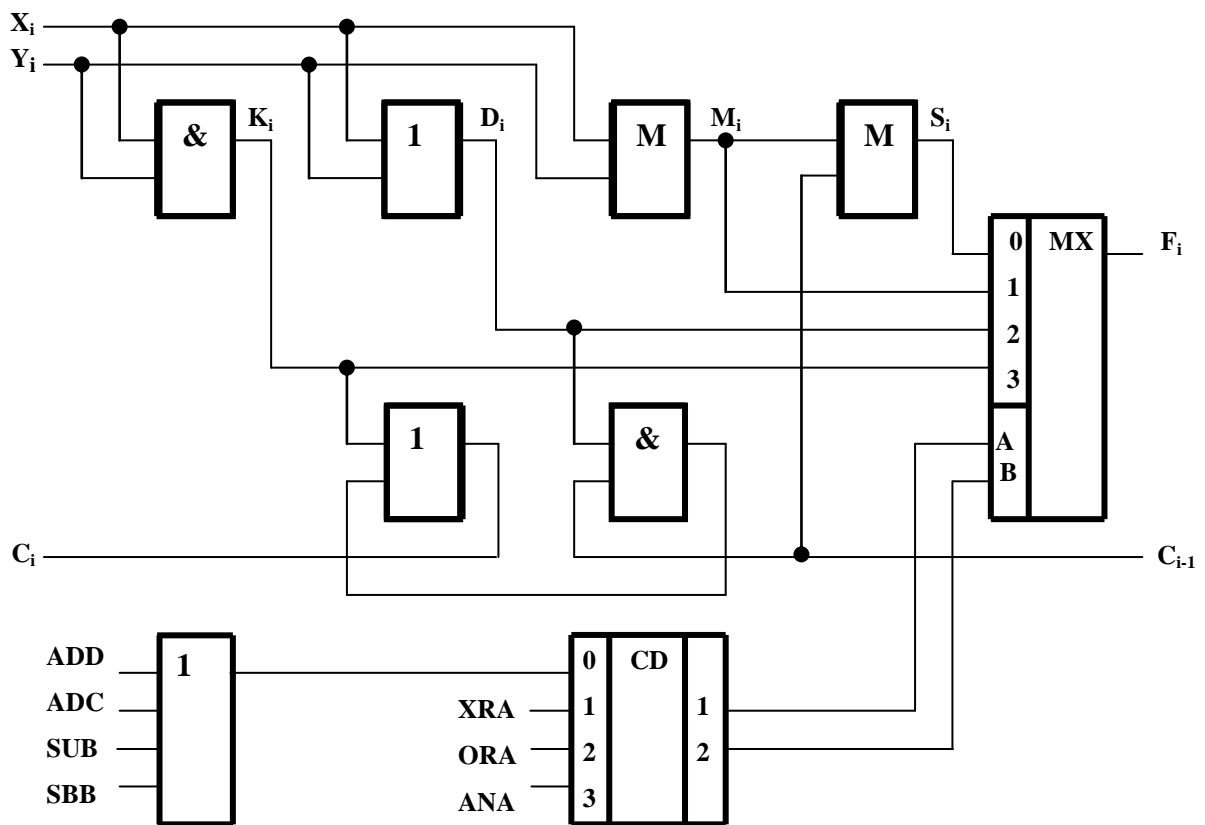


Рисунок 9 - Функціональна схема розряду АЛП мікропроцесора i8080

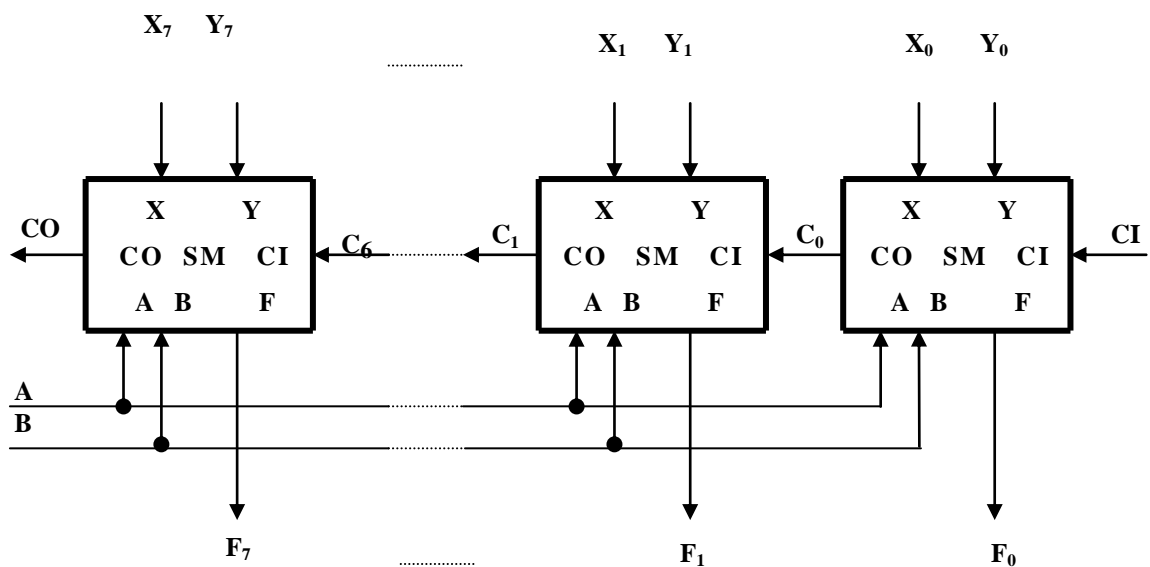


Рисунок 10- Організація паралельного операційного елементу мікропроцесора

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

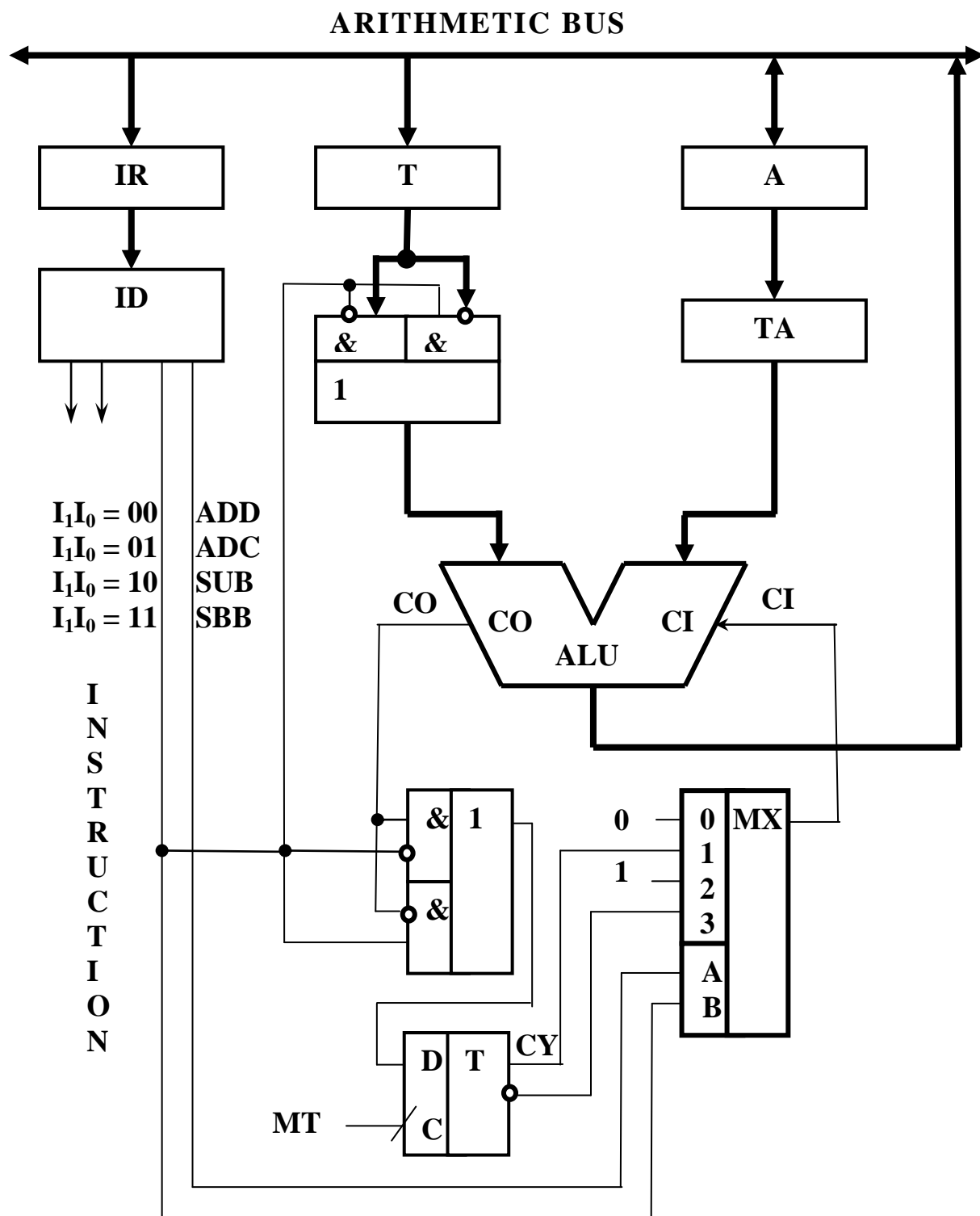
26

Як видно зі схеми, перенесення з молодшого розряду C_{i-1} впливає на функцію суми S_i та перенесення в старший розряд C_i . Логічна схема формування перенесення в старший розряд не складна, а на малюнку вона побудована на двох логічних елементах з урахуванням сформованих базових функцій K_i , D_i . На вищеназвані три основні логічні функції вхідне перенесення не впливає. В реальній схемі мікропроцесора зосереджена допоміжна логіка паралельного розповсюдження перенесення, що не перешкоджає цьому виконанню в інтегральній технології, але підвищує швидкодію роботи чотирьох функційного суматора.

Чотири однорозрядні суматори поєднані між собою в групу. В свою чергу, подібні дві групи з послідовним між груповим перенесенням та паралельним внутрішнім груповим складають восьми розрядний операційний пристрій суматора мікропроцесора. Це визначено, в першу чергу, з необхідністю обчислень в мікропроцесорі двійково – десяткових чисел і формування в зв'язку з цим арифметичного прапора допоміжного перенесення. Керуючі входи A , B кожної однорозрядної схеми поєднані між собою, а сигнали управління по ним, як раніше було показано, поступають від пристрою декодування інструкцій системи команд мікропроцесора. Вхідне перенесення (CI – *Carry Input*) формується виключно згідно виконуємої команди, вихідне перенесення (CO – *Carry Output*) з попередньою ознакою фіксується з метою подальшого використання, ураховуючи попередні обчислювальні дії мікропроцесора. Згідно з думкою автора цього творчого утворення таку структуру будови операційного пристрою мають більш потужні сучасні процесори.

Арифметико логічний пристрій (АЛП) мікропроцесора акумуляторного типу передбачає присутність в якості основного регістру так званого акумулятора, зберігаючого перший операнд до виконання операції в мікропроцесорі і фіксуючого результат. Без сумнівів, це спрощує склад обчислювального пристрою мікропроцесора. Показане на малюнку оточення комбінаційної логіки суматора допоміжними регістрами і функціональними елементами складає спрощену структурну схему АЛП мікропроцесора при арифметичних обчисленнях.

					ДП 5.05010201 435 01 ПЗ	Арк.
						27
Змін.	Арк.	№ докум.	Підпис	Дата		



Акумулятор (*A* – *Accumulator*)

Рисунок 11 - Структурна схема АЛП мікропроцесора i8080 при виконанні інструкцій складання і віднімання

має другу ступінь тимчасової пам'яті (*TA -Time Accumulator*) з метою збереження в головному накопичувачі результату обчислення. Регістр тимчасового збереження другого операнду (*T - Time register*) сприймає дані від програмно адресованого

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

28

командою елементу для обчислень в АЛП та передає через допоміжну логіку в комбінаційну частину значення прямим або оберненим виразом. Між вихідним перенесенням суматора та вхідним знаходиться оригінальна логічна схема. Мультиплексор формує перенесення входу, залежно від дії арифметичного текучого обчислення та згідно з правилом формули його роботи

$$CI = (\overline{B} \cdot \overline{A}) \& 0 \vee (\overline{B} \cdot A) \& CY \vee (B \cdot \overline{A}) \& 1 \vee (B \cdot A) \& \neg CY$$

Вхідне перенесення буде дорівнювати нулю при виконанні складання без урахування прапора перенесення CY та значенню прапора CY з урахуванням, що витікає з самого ствердження. З відніманням пов'язані інші дії. Оскільки від'ємник замінюється доповненням для його складання зі зменшуваним, то обернене його значення на входах суматора з плюсом одиницею перенесення на вході і є те доповненням другого операнду. Урахування позики при відніманні подібне відніманню без позики, але ще необхідно віднімати стан прапора CY . Операція

$$I-CY = \text{not}(CY)$$

є запереченням прапора перенесення. Тому в цьому випадку вхідне перенесення дорівнює оберненому значенню прапора основного перенесення.

Відносно вихідного перенесення, сформованого після додавання чи віднімання, то дії більш спрощені при його запам'ятовуванні і збереженні в тригері перенесення, виконуючого функцію формувача прапора CY . Функція входу D тригера визначається керуванням I_1 від декодера інструкції та значенням перенесення на виході CO і має такий аналітичний вираз

$$D = \text{not}(I_1) \text{ and } (CO) \text{ or } (I_1) \text{ and } \text{not}(CO).$$

Це свідчить, що після складання прапор переносу CY дорівнює прямому значенню вихідного переносу CO , а після віднімання – оберненому. Усі ці дії при роботі дуже швидкісні та приховані від користувача і вперше висвітлені у цій монографії (Примітка автора).

					ДП 5.05010201 435 01 ПЗ	Арк.
						29
Змін.	Арк.	№ докум.	Підпис	Дата		

Отриманий із програмної пам'яті код команди в регістр команд мікропроцесора (**IR – Instruction Register**) декодується дешифратором (**ID – Instruction Decoder**), чим забезпечується на текучому командному циклі коректне керування АЛП зі сторони пристрою управління мікропроцесором.

В якості приклада корисно розглянути хід виконання команди в мікропроцесорі віднімання

SUB D; A:= (A) – (D)

Пристрій керування сформує управління $I_1 I_0 = 10$, що приведе до такого стану. Вхідне перенесення $CI = 1$, а регістр T отримає код $7A h$. В суматор поступає обернене значення регістра T , рівне $85 h$. Буфер акумулятора зафіксує число $2C h$.

Після виконання команди стан акумулятора $(A) = B2 h$, а стан прапора $(CY) = 1$, що свідчить про позику.

Стан регістрів до виконання

$(A) = 2C h; (D) = 7A h.$

Стан регістрів при виконанні

$(A) = 2C h; (TA) = 2C h; (T) = 85 h.$

Стан суматора пристрою

0	1
↑	↑
0010 1100	
+ 1000 0101	
1011 0010	

Вихідне CO =0 Вхідне CI =1

Перенесення

Команда віднімання чотирьох тактна, але за рахунок двохфазного внутрішнього тактування вона виконається за один стан (такт) головного мікропроцесора.

Завершуючи висвітлення теми матеріалу, можна відмітити, що усі на перший погляд складні інформаційно логічні дії виконуються досить просто в типових операційних пристроях за рахунок програмно керованого управління. Зміна стану елементів АЛП мікропроцесора автоматично синхронізується зі зміною виконуємої інструкції, що є однією із ознак принципу програмного керування.

1.2.2 Пристрій виконання десяткової корекції

Пристрій арифметико – логічної обробки головного процесора розраховано на числову програмну обробку двійкових цілих чисел. Передбачена можливість часткової

					ДП 5.05010201 435 01 ПЗ	Арк.
						30
Змін.	Арк.	№ докум.	Підпис	Дата		

обробки в мікропроцесорі двійково – десяткових чисел у форматі в один байт, що відповідає зображенню двох десяткових розрядів. Більш точніше, ця обробка полягає у виконанні безоперандної команди десяткової корекції після додавання упакованих в один байт десяткових чисел

DAA (Decimal Adjust after Addition).

Команда тестує в два етапи стан акумулятора процесора на ознаку переповнення розрядної сітки. Спочатку перевіряється молодша потім старша тетрада, а у випадку переповнення виконується корекція однієї із тетрад або по черзі обох. Алгоритм роботи команди має такий операційний вигляд

If ((A) and 0F h) > 09 or ((AC) =1) then

A := (A) - 0A h + 10 h = (A) + 06; AC := 1.

If (A) > 9F or ((CY) =1) then

A := (A) - A0 h + 100 h = (A) + 60; CY := 1.

Ознакою переповнення після додавання є результат більший числа дев'ять або було перенесення в старший розряд. Корекція полягає в відніманні від значення числа десять та додавання одиниці в старшу тетраду, що після узагальнення відповідає додаванню до переповненого значення числа шість. Факт корекції відмічається установкою відповідного прапора перенесення у одиницю.

Команда корекції активує роботу вмонтованого в АЛП десяткового коректора, збудованого всередині мікропроцесора на логічних елементах. Синтез схеми коректора виконано по рівнянням, отриманих з мінімізуючих карт Карно.

					<i>ДП 5.05010201 435 01 ПЗ</i>	Арк.
						31
Змін.	Арк.	№ докум.	Підпис	Дата		

D1D0

KL

00

01

11

10

00

0

0

0

0

01

0

0

0

0

11

1

1

1

1

10

0

0

1

1

D3D2

D5D4

KH

00

01

11

10

00

0

0

0

0

01

0

0

0

0

11

1

1

1

1

10

0

0

1

1

D7D6

Рисунок 12 - Десяткова корекція після додавання запакованих десяткових чисел

Функції корекції молодшої **Kl** та старшої **Kh** тетради подібні з незначним відрізненням за рахунок використання відповідних розрядів акумулятора та прапорів перенесення. Після склеювань в клітинах карти та перетворень визначаються такі рівняння

$$\begin{aligned}
 Kl &= D3 \& D2 \vee D3 \& D1 \vee (AC); \\
 Kh &= D7 \& D6 \vee D7 \& D5 \vee (CY); \\
 K &= \overline{M} \& Kl \vee M \& Kh = \\
 &= \overline{M} \& [D3 \& D2 \vee D3 \& D1 \vee (AC)] \vee \\
 &\vee M \& [D7 \& D6 \vee D7 \& D5 \vee (CY)].
 \end{aligned}$$

Функція загальної корекції **K** поєднує функції корекції молодшої **Kl** та старшої **Kh** тетрад за рахунок керуючого внутрішнього сигналу **M**. При **M=0** корегується молодша тетрада, а при **M=1** – старша. Керуючий сигнал **M** впливає на вибір потрібної тетради акумулятора при її корекції шляхом формування та додавання числа шість. Він також забезпечує селекцію допоміжної логіки керування записом прапора перенесення **AC** при обробці молодшої тетради і прапора **CY** при корекції старшої тетради, як це видно зі схеми.

Схема корекції реалізована з використанням мультиплексора, комбінаційного формувача, демультимплексора. Вона підключена до відповідних виходів акумулятора і ураховує попередній стан прапорів допоміжного і основного перенесень, як це раніше описано логічним рівнянням. Навмисно пристрій суматора поділено на молодшу і старшу групи з показом вихідних перенесень від кожної. На часовому такті ***T3*** машинного циклу команди корекції в пристрої коректується молодша тетрада, а на такті ***T4*** – старша.

На третьому такті керуючий сигнал ***M = 0***, а це є умовою вибіру верхніх частин мультиплексора, демультимплексора і молодшої тетради суматора. У випадку виконання умови переповнення молодшої тетради в акумуляторі функція виходу ***K=1*** через демультимплексор в якості другого операнду в суматор сформує код корекції ***06 H***. Вмісті з цим спрацює логіка керування запису в тригер допоміжного переносу, а його стан буде дорівнювати ***AC=1***. Якщо ***K=0***, процес додавання блокується, корекція не виконується, прапор ***AC*** не змінюється.

Подібні дії виконуються над старшою тетрадою. При виявленні переповнення склад акумулятора коректується додаванням числа ***60 H*** та записом одиниці в тригер основного перенесення ***CY := 1***. Такі обчислювальні дії забезпечуються на останньому четвертому такті циклу за рахунок сформованого керуючого сигналу ***M = 1***.

Логіка керування тригерами збереження прапорів перенесень ураховує їх модифікацію не тільки функцією загальної корекції ***K***, але і вихідними перенесеннями від відповідних груп суматора. Тому при не виконанні корекції тетрад акумулятора (переповнення не було після додавання десяткових чисел) стан прапорів визначається виключно станом вихідних перенесень ***C3, C7***.

При роботі команди корекції ***DAА*** регістр тимчасового збереження другого операнду ***T*** скидається, а в роботу вступає диз'юнктор, який пропускає від схеми корекції коди ***00 H, 06 H, 60 H***, розподілені в часу тактів машинного циклу ***T3, T4***. Завжди після корекції відповідної тетради отриманий результат записується в акумулятор, а регістрі прапорів, крім вище названих двох, фіксує ще три прапорці стану текучої програми.

					ДП 5.05010201 435 01 ПЗ	Арк.
						34
Змін.	Арк.	№ докум.	Підпис	Дата		

1.3 Будова інтерфейс оперативної пам'яті з модулем процесора

Оперативна пам'ять відноситься до двонаправленої тому, що допускає як запис даних, так і їх читання. У спрощених мікропроцесорних системах оперативна пам'ять використовується для збереження як пам'яті даних, так програмної та системного стеку. Системна магістраль формується за рахунок використання допоміжних мікросхем. Так магістраль адреси **BA15-BA0** передає через буфер **BA** активну адресу протягом усього машинного циклу. Елементи пам'яті для будови буферу не використовуються тому, що локальна шина адреси головного мікропроцесора ізольована від інших виводів і зберігає протягом усього машинного циклу адресу. Буфер даних **BD** двонаправлений, на відміну від буфера адреси, отримує від головного процесора сигнали управління вибору буферу та керування напрямком передачі даних. Розрядність буферу складає вісім біт **BD7-BD0**. Через буфер процесор отримує команди від програмної пам'яті, дані - від пам'яті даних та від стеку або портів введення. При виведенні в пам'ять даних або порти виводу буфер прозорий від головного процесора в зовнішнє середовище.

Формування командних сигналів читання \overline{MRDC} та запису \overline{MWTC} фізичної пам'яті виконується з урахуванням певних ознак слова стану процесора та відповідних стробів, які процесор залежно від напрямку передачі байту відповідно формує. Логіка формування сигналів управління визначена рівняннями та відповідною схемою

$$MRDC = MEMR \& DBIN;$$

$$\overline{MRDC} = \overline{MEMR} \& \overline{DBIN} ;$$

$$MWTC = \overline{OUT} \& WR;$$

$$\overline{MWTC} = \overline{\overline{OUT} \& WR} = OUT \vee \overline{WR}.$$

Дешифратор адреси вибирається у випадку, якщо процесор пише або читає пам'ять. Формування вибору \overline{CS}_{DC} пам'яті виконується клапаном кон'юнкції

$$CS_{DC} = MRDC \vee MWTC;$$

$$\overline{CS}_{DC} = \overline{MRDC} \& \overline{MWTC}.$$

					ДП 5.05010201 435 01 ПЗ	Арк.
						35
Змін.	Арк.	№ докум.	Підпис	Дата		

Ураховуючи вище описаний принцип будови, спрощена схема підключення модуля пам'яті до системної магістралі має такий вигляд.

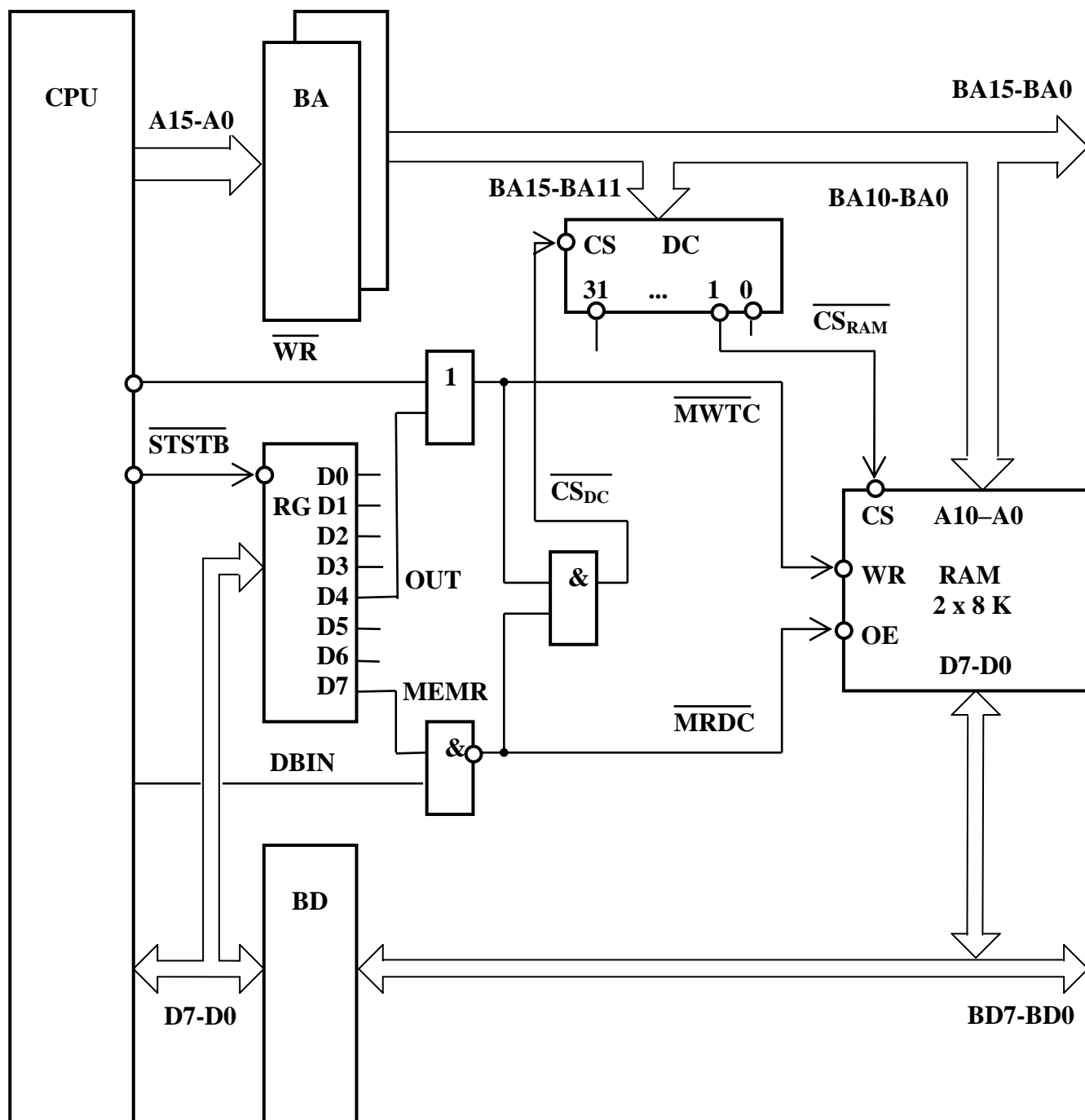


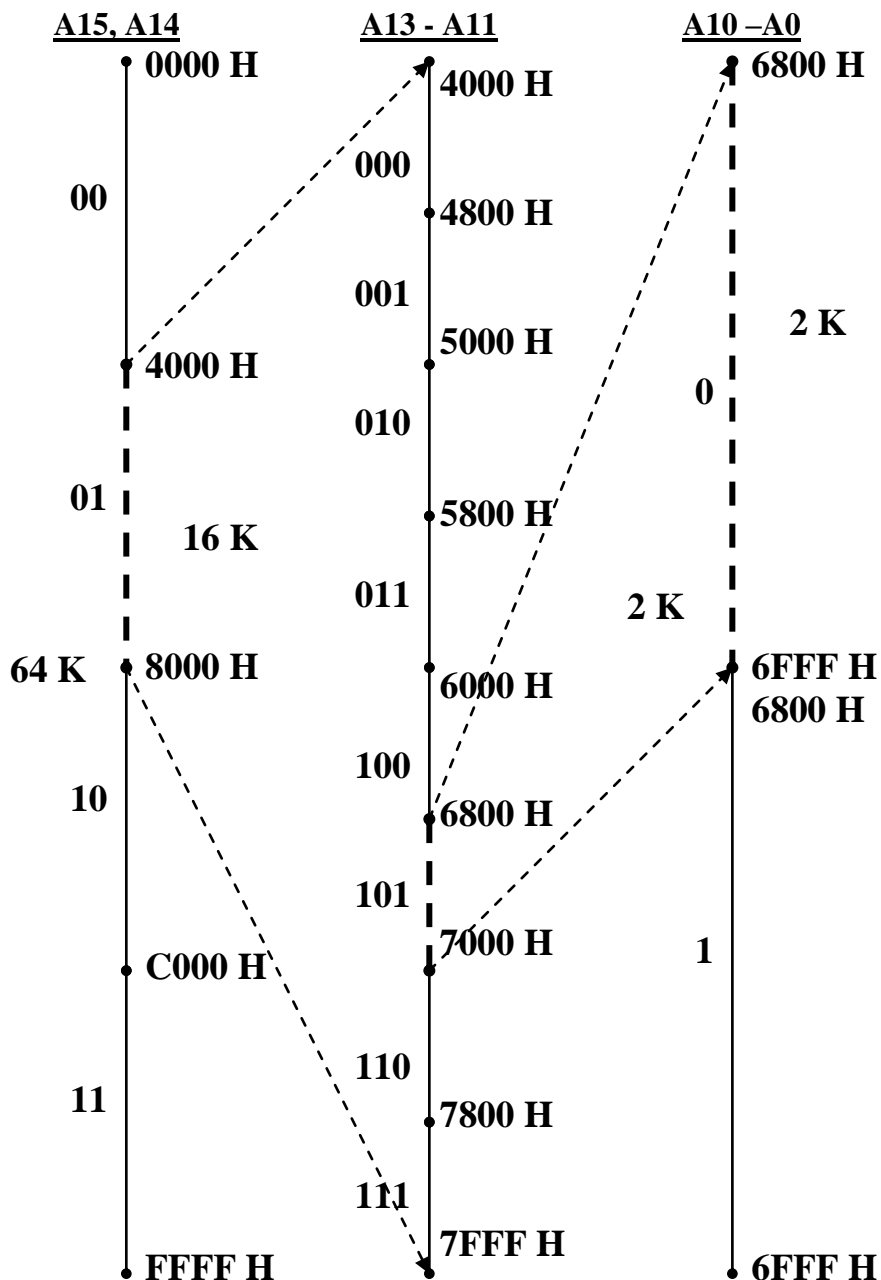
Рисунок 14 - Інтерфейс оперативної пам'яті з модулем центрального процесора

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

36



$$\begin{aligned}
 CS_{RAM} &= (\overline{CS_{CD}} \& \overline{A15} \& \overline{A14}) \& (A13 \& \overline{A12} \& A11) = \\
 &= (\overline{CS_{CD}} \& \overline{A15} \& \overline{A14}) \& F5; \\
 \overline{CS_{RAM}} &= \overline{(\overline{CS_{CD}} \& \overline{A15} \& \overline{A14}) \& F5}.
 \end{aligned}$$

Рисунок 15 – Карта пам'яті розміщення мікросхеми оперативної пам'яті в фізичному просторі та логіка керування дешифратором вибору

Змін.	Арк.	№ докум.	Підпис	Дата

ДП 5.05010201 435 01 ПЗ

Арк.

37

На верхньому малюнку показана карта пам'яті розміщення мікросхеми пам'яті за адресою **6800 H** ємністю два кілобайти. За допомогою карти отримані логічні рівняння для функції вибору мікросхеми пам'яті в допущенні використання мікросхем середнього ступеню інтегрованості. Ураховуючи попередні тлумачення відносно формування командних сигналів \overline{MRDC} і \overline{MWTC} та отримані рівняння, унизу побудована спрощена схема підключення мікросхеми пам'яті до системної магістралі мікросистеми.

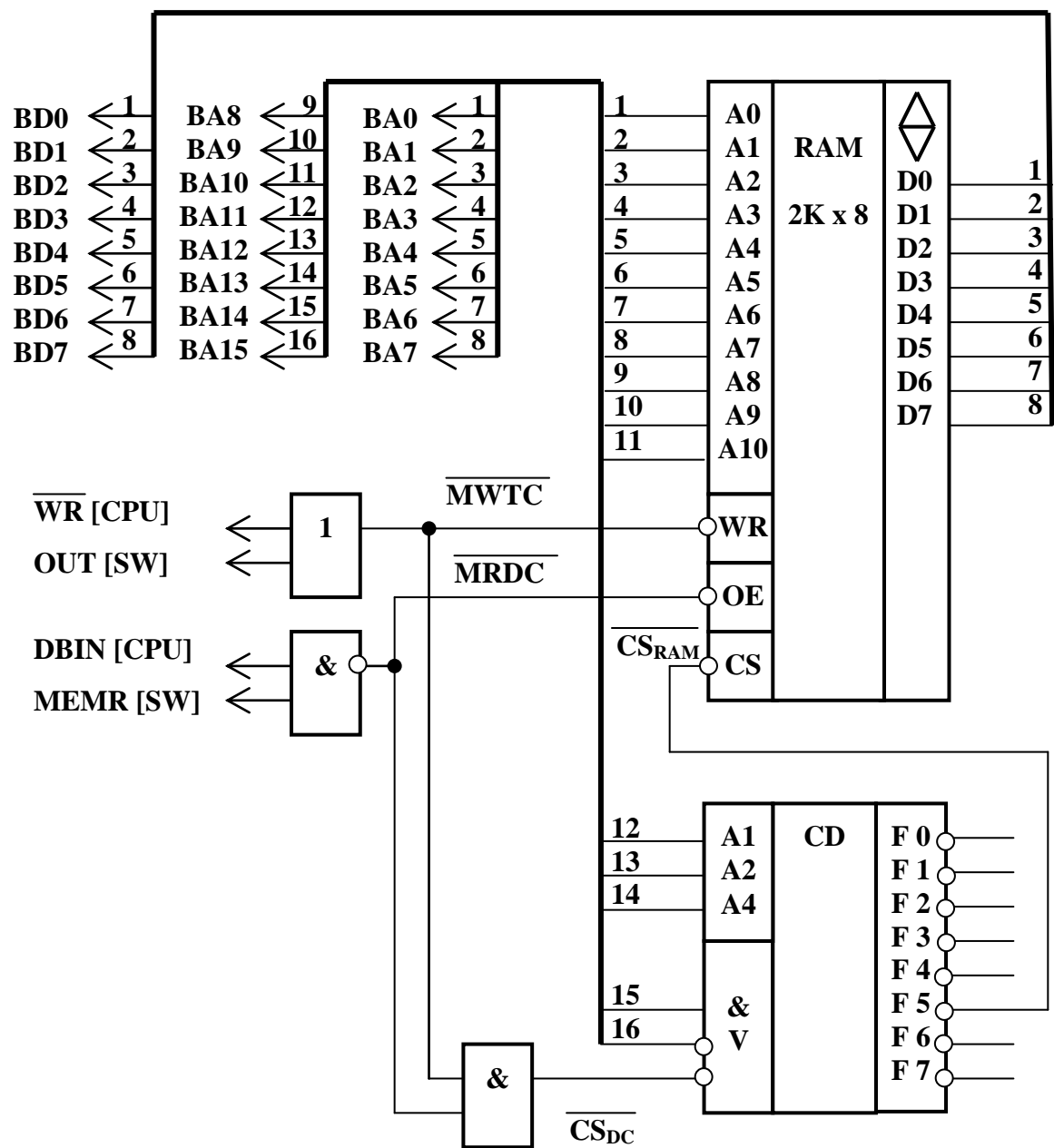


Рисунок 16– Фрагмент схеми інтерфейсу пам'яті з системною магістраллю мікропроцесорної системи