

Міністерство освіти і науки України
Державний університет телекомунікацій
Навчально-науковий інститут Телекомунікацій та інформатизації

Кафедра Комутаційних систем

КЕРУВАЛЬНІ КОМПЛЕКСИ ЕЛЕКТРОЗВ'ЯЗКУ

Лекція 4

Київ – 2014

Розробник _____

**к.т.н., доц. каф.КС
К.П.Сторчак**

ЗАТВЕРДЖУЮ

Зав.каф. _____ В.І. Гостєв

”__” _____ 2014 року

Тема 4. ПРОЦЕСОРИ ЕКМ. СПОСОБИ АДРЕСАЦІЇ ДАНИХ В ЕКМ. ФОРМАТИ ПРЕДСТАВЛЕННЯ ЧИСЕЛ В ЦИФРОВИХ ПРОЦЕСОРАХ

План:

- 1 Структура процесора.
 - 2 Способи адресації. Алгоритм роботи процесора.
 - 3 Формати представлення чисел в цифрових процесорах.
- Завдання на СРС
1. Поглиблене вивчення основних положень лекції.
 2. Підготовка до лабораторної роботи.

1 Структура процесора

Схема процесора приведена на рисунку 1. Процесор складається з блока регістрів БР, арифметично-логічного пристрою АЛУ, блоку узгодження з інтерфейсом БС блоку мікропрограмного управління БМУ.

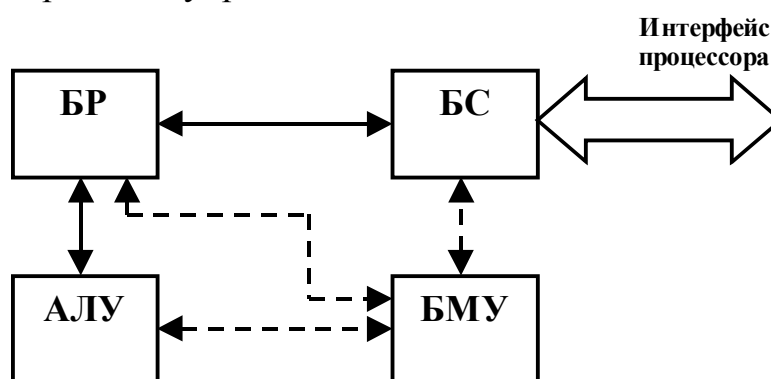


Рисунок 1 - Структура процесора

У блоці регістрів БР всі регістри можуть бути розділені на програмно-доступні й програмно-недоступні.

До програмно-доступних регістрів відносяться *регістри*, вміст яких доступний по командах процесора, тобто може використовуватись в операціях, змінюватися по операціям, що задаються командами. Вміст цих регістрів не змінюється від команди до команди. До таких регістрів відносяться акумулятор; регістри базові й індексні; множника-частки; покажчика стеків, лічильник команд і т.п. У сучасних процесорах програмно-доступні регістри звичайно не закріплюють жорстко по призначенню (за винятком лічильника команд і покажчика стека), а надають програмістові кілька регістрів загального призначення РОН, які він може використати в якості накопичуючих, базових, індексних і т.д. Кількість таких регістрів може бути від 8 до декількох десятків (30-75).

Програмно-недоступні регістри є робочими регістрами процесора й призначені для зберігання різноманітної інформації в процесі виконання однієї команди. До таких регістрів відносяться в першу чергу регістр команди (РК), буферні регістри адрес (РАП) і слів (РСП), що використовуються при звертанні до пам'яті, і інші регістри.

Арифметично-логічний пристрій (АЛУ) призначений для виконання арифметичних і логічних операцій над словами даних і адресами. В АЛУ виробляються й зберігаються признаки результатів. У процесорах ЕОМ загального призначення АЛУ має складну структуру й містить у собі блоки, що обробляють числа з фіксованою крапкою різної довжини, числа із плаваючою крапкою, десятково-кодовані числа, поля змінної довжини. Для обчислення адрес (при індексній і відносній адресації) іноді використовується окремий АЛУ адрес.

У міні-ЕОМ в основному АЛУ виконуються арифметичні операції над числами з фіксованою крапкою й логічні операції. Передбачається можливість підключення додаткового АЛУ, що виконує операції із плаваючою крапкою й інші операції розширеної арифметики.

Іноді мікроЕОМ додаткові АЛУ не включаються до складу процесора, а підключаються до інтерфейсу як периферійні пристрої (ПУ).

Блок узгодження з інтерфейсом (БС) забезпечує підключення шин інтерфейсу й утворення всіх необхідних сигналів для виконання обміну по шинах. До складу БС звичайно включають контролер шин всієї ЕОМ.

Інтерфейс процесора містить у собі шини адреси, даних (вхідні й вихідні або двоспрямовані) і керування. Якщо використовуються ті самі фізичні шини для передачі адрес і даних, то передбачаються сигнали ідентифікації. До складу шин керування входять шини, що задають операцію вводу або виводу (щодо процесора), і шина синхронізації.

2 Способи адресації. Алгоритм роботи процесора

Спосіб адресації - спосіб обчислення адреси операнду на основі інформації в команді. Спочатку в ЕОМ використовувався тільки один спосіб адресації - пряма адресація. Недоліки цього способу викликали необхідність використання інших способів.

У системі команд процесора передбачається кілька способів адресації операндів для однієї й тієї ж операції. Тому в адресній частині команди є спеціальне поле - *поле признаків операції*, у якому кодується признак адресації. Якщо такого признака немає, то спосіб адресації визначається по коду операції. При використанні поля признаків адресації будь-яка операція може виконуватися з будь-якими способами адресації, що спрощує програмування.

Розглянемо найпоширеніші способи адресації даних, що застосовуються в сучасних процесорах.

Неявна адресація. Цей спосіб адресації є у всіх процесорах. Основне призначення його - зменшення довжини команди за рахунок виключення частини адрес. Кількість фактичних адрес вдається зменшити за рахунок використання неявної адресації. При неявній адресації адреса обчислюється по будь-якій формулі з наявних у команді відомостей або в якості операнда використовується вміст фіксованої комірки пам'яті або регістра процесора. Перший спосіб застосовується, наприклад, для адреси наступної команди, що обчислюється додаванням деякого числа l до адреси виконуваної команди (l - це довжина команди в мінімальних адресуємих елементах

інформації в пам'яті). Довжина команди в більшості сучасних процесорів число змінне й залежить від типу команди.

Аналіз програм показує, що необхідно в якості одного з операндів використати результат попередньої операції. Тому в якості адреси результату звичайно використовується адреса одного з операндів і результат заміщає один з операндов. Це дозволяє значно скоротити довжину адресної частини команди, але приводить до необхідності введення посилкових операцій, які не виконують ніяких дій, крім копіювання інформації в пам'яті або між пам'яттю й одним з реєстрів процесора.

Якщо в якості операнда використати вміст одного з реєстрів процесора й туди ж записувати результат операції то можна обійтися й одною, фактично присутньою у команді адресою. Регістр процесора, вміст якого використовується в операціях і куди записується результат операції називається *накопичувальним реєстром* або *акумулятором*.

Залежно від кількості фактично присутніх у команді адрес вона називається *безадресною, одноадресною, двоадресною* і т.д. У сучасних процесорах звичайно застосовуються не більш ніж двоадресні команди, причому адресність команд є змінною величиною й саме вона звичайно впливає на розмір команди. Так, для унарних операцій (порозрядна інверсія, зміна знака й ін.) потрібно завжди на одну адресу менше, ніж для бінарних; команди переходів вимагають вказівки однієї адреси, деякі команди, що змінюють стан процесора, взагалі не вимагають операндів.

Безпосередня адресація. У програмах дуже часто необхідно виконувати операції, у яких один з операндів є константою, що не змінюється в процесі роботи програми. Можна записати таку константу в комірку пам'яті й адресувати її звичайним чином, але краще для цієї мети використати безпосередню адресацію (операнд при цьому записується безпосередньо в команді замість однієї з адрес). Такий спосіб адресації прискорює виконання команди, тому що не потрібно циклу звертання до пам'яті за операндом (він обраний разом з командою), і одночасно заощаджує пам'ять.

Пряма адресація. Вказівка в команді прямої адреси операнда, тобто номера комірки пам'яті, у якій він записаний, - простий і досить універсальний спосіб адресації, широко використовуваний і в ЕОМ першого й другого покоління. У цей час він знаходить обмежене використання через труднощі:

- переміщення програм у пам'яті;
- обробки масивів даних і організації спеціальних видів пам'яті;
- обробки даних, організованих в облікові структури;
- передачі фактичних параметрів у підпрограми.

У всіх цих випадках необхідні модифікація або формування адресних частин команд, що ускладнює програмування й самі програми.

Непряма адресація. Адреса операнда, що зберігається у деякій комірці пам'яті, називається *показчиком*; адресація до операнду через ланцюжок показчиків - *непрямою адресацією*; кількість показчиків у ланцюжку - *кратність непрямої адресації*. Найпоширенішою однократною непрямою адресацією, схема якої наведена на рисунку 7.2. Значення показчика, тобто адреса операнда, називається *непрямою адресою*. Адреса показчика залишається постійним, а непряма адреса може змінюватися в процесі виконання програми командами самої програми. Цим

забезпечується можливість переадресації даних, тобто обробка масивів операндів або складної структури даних за допомогою однієї й тієї ж програми або ділянок програми.

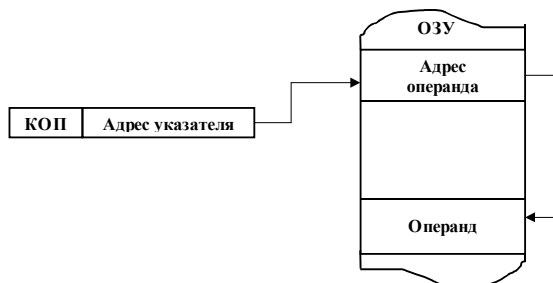


Рисунок 2 - Схема непрямой адресації

Непряма адресація спрощує обробку масивів і спискових структур даних, а також передачу параметрів підпрограмам, але не забезпечує переміщення програм у пам'яті.

При багаторазовій непрямої адресації в команді вказується адреса першого покажчика й кратність - число послідовно обраних покажчиків. Значення кожного покажчика - адреса наступного покажчика. В останньому покажчику знаходиться адреса операнда. На практиці кратність звичайно не перевищує 6-8, тому що при кратності >2 програмування значно утрудняється.

Пряма адресація може розглядатися як окремий випадок непрямої адресації із кратністю 0.

Команда з непрямою адресацією виконує додаткові звертання до пам'яті за непрямою адресою й виконується довше, ніж команда із прямою адресацією. Але застосування команд із непрямою адресацією вигідніше, тому що аналогічні дії з використанням тільки прямої адресації вимагають виконання цілої програми.

Індексна адресація. Якщо в якості операндів використовуються змінні з індексами, то вони позначають елементи масивів, що представляють собою сукупність однотипних значень. Найбільш зручна для роботи з масивами індексна адресація, схема якої наведена на рисунку 3. При індексній адресації використовуються індексні реєстри, розташовані в процесорі.

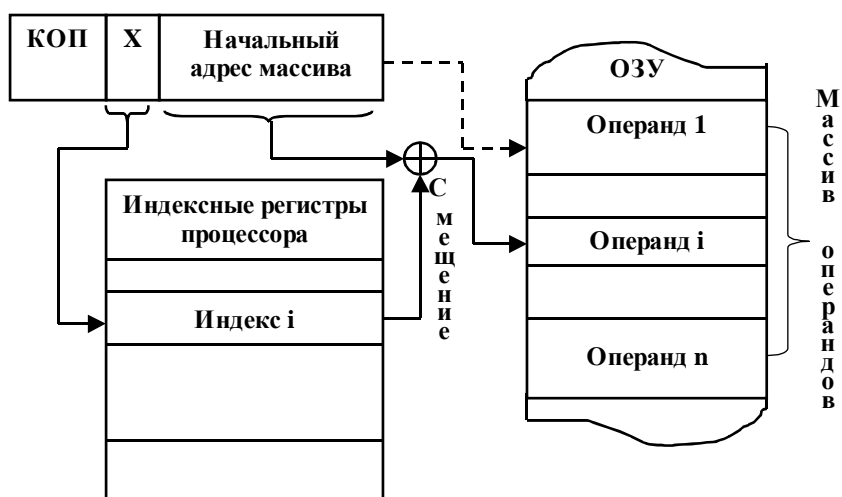


Рисунок 3 - Схема индексной адресації

команди виклику підпрограми, що запам'ятовує вміст лічильника команд в одному з реєстрів процесора.

В якості *базового реєстра* можна використати й *лічильник команд*. У цьому випадку не потрібно відводити окремий реєстр для зберігання базової адреси й не потрібно робити завантаження його. Недолік адресації щодо лічильника команд у тому, що при використанні того самого операнда в різних командах потрібно вказувати різну величину зміщення.

Автоіндексна адресація. У міні- і мікроЕОМ, де обладнання процесора обмежене, застосовується автоіндексна адресація, що є продовженням непрямої адресації. При автоіндексній адресації непряма адреса збільшується або зменшується до або після виконання операції з операндом. Цей спосіб адресації забезпечує високу ефективність обробки масивів при невеликих апаратурних витратах на його реалізацію, особливо у випадку зберігання непрямої адреси на внутрішньому реєстрі процесора. Якщо непряма адреса збільшується, то адресація називається *автоінкрементною* або з *автозбільшенням*, якщо зменшується - то *автодекрементною* або з *автозменшенням*. Звичайно ці способи узгоджені таким чином, щоб при послідовному застосуванні, наприклад, автодекрементної і автоінкрементної адресації використовувався той самий операнд. Це вимагає, щоб у розглянутому випадку при автодекрементній адресації спочатку відбувалося зменшення непрямої адреси, а потім операція, а при автоінкрементній адресації навпаки - спочатку операція з операндом, а потім збільшення непрямої адреси.

Стекова пам'ять і адресація до неї. Стекова пам'ять відноситься до спеціальних видів пам'яті. При звертанні до неї адреса не використовується. Для зчитування/запису доступна тільки одна комірка, що називається *вершиною стека*.

На рисунку 5 показаний стан стекової пам'яті при послідовному записі в стек змінних *a* й *b* і зчитуванні їх зі стека. Як видно з рисунка, останній записаний елемент буде прочитаний першим. Подібний порядок запису/зчитування характерний для адрес повернення при вкладених викликах підпрограм, а також у багатьох інших випадках.

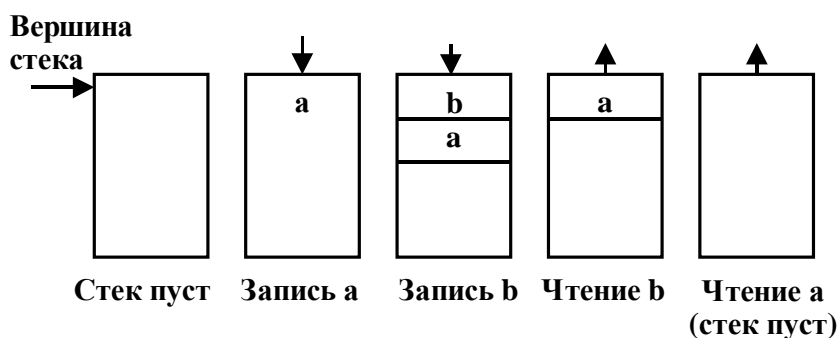


Рисунок 5 - Функціонування стекової пам'яті

Практично стекова пам'ять реалізується на основі звичайної пам'яті з використанням покажчика стека й автоіндексної адресації. Запис у стек здійснюється з використанням автодекрементної адресації, а зчитування - автоінкрементною. На рисунку 6 наведений послідовний стан покажчика стека й комірок пам'яті при послідовному записі в нього чисел 5, 12, 8 і зчитуванні числа 8. Передбачається

описаний вище порядок виконання автоінкрементної і автодекрементної адресації. Операції зчитування/запису зі стековою пам'яттю виконуються процесором неявно при викликах підпрограм і поверненнях з підпрограм, при обробці переривань і в інших випадках. У цих операціях передбачене занесення до стеку вмісту тільки деяких регістрів (акумулятора, регістра ознак). Для програмування в системі команд процесора звичайно передбачаються операції пересилання значень зі стека у пам'ять або регістр загального призначення (РОН) і в стек з пам'яті або регістрів загального призначення й операції завантаження покажчика стека, що мають назву ініціалізація покажчика стека.

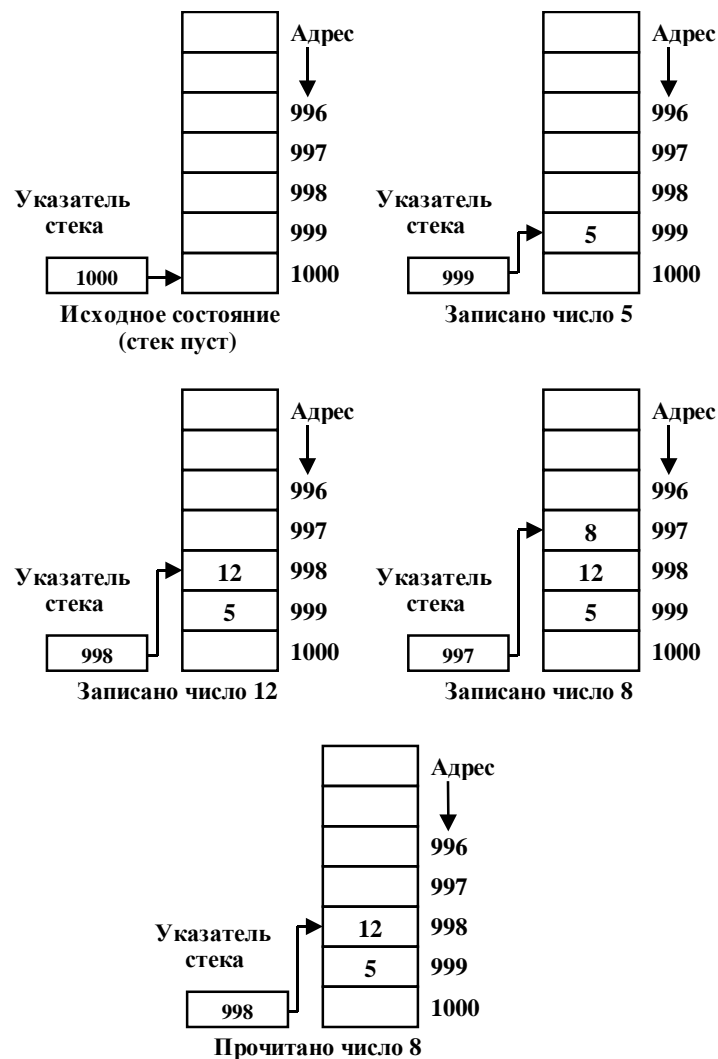


Рисунок 6 - Функціонування стекової пам'яті з покажчиком стека

