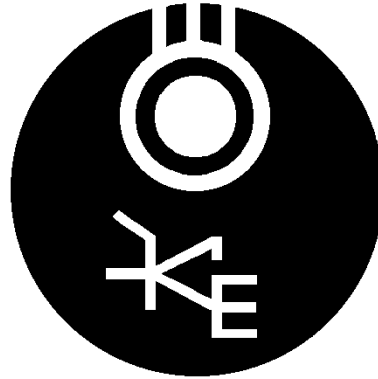


П.І.Б. _____
Група _____
Варіант _____
Відмітка про залік:

Міністерство освіти і науки України
Національний гірничий університет



МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи МПП-1

“АРХІТЕКТУРА МІКРОКОНТРОЛЕРА 8051АН СІМЕЙСТВА MCS-51.

ПОВНОЕКРАННИЙ НАЛАГОДЖУВАЧ-СИМУЛЯТОР FD51”,

індивідуальних завдань та самостійної роботи

з професійно-орієнтованої дисципліни

“МІКРОПРОЦЕСОРНІ ПРИСТРОЇ”

для студентів спеціальності 7.092203 “Електромеханічні системи автоматизації
та електропривод” напряму “Електромеханіка”

Дніпропетровськ

2011

Міністерство освіти і науки України
Національний гірничий університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи МПП-1 “Архітектура мікроконтролера 8051АН сімейства MCS-51. Повноекранний налагоджувач-симулятор FD51”, індивідуальних завдань та самостійної роботи з професійно-орієнтованої дисципліни “Мікропроцесорні пристрої” для студентів спеціальності 7.092203 “Електромеханічні системи автоматизації та електропривод” напряму “Електромеханіка”

Дніпропетровськ

НГУ

2011

Методичні вказівки до виконання лабораторної роботи МПП-1 “Архітектура мікроконтролера 8051АН сімейства MCS-51. Повноекранний налагоджувач-симулятор FD51”, індивідуальних завдань та самостійної роботи з професійно-орієнтованої дисципліни “Мікропроцесорні пристрої” для студентів спеціальності 7.092203 “Електромеханічні системи автоматизації та електропривод” напрямку “Електромеханіка”/ Упорядн.: В. І. Кириченко, О. А. Яланський. – Дніпропетровськ: Національний гірничий університет, 2011. – 41 с.

Упорядники: В. І. Кириченко, д-р техн. наук, проф.

О. А. Яланський, канд. техн. наук, доц.

Відповідальний за випуск завідувач кафедри електропривода

О. С. Бешта, д-р техн. наук, проф.

ЗМІСТ

Вступ.....	5
1. Зміст самостійної та лабораторної робіт.....	5
1.1. Самостійна робота.....	5
1.2. Лабораторна робота.....	5
2. Історія створення та розвитку мікропроцесорів і мікроконтролерів.....	6
3. Структурна організація мікроконтролерів сімейства MCS-51.....	11
3.1. Загальні відомості про архітектуру мікроконтролера.....	11
3.2. Організація ОЗП, ПЗП та регістрів мікроконтролера.....	15
Запитання для самоперевірки.....	19
4. Повноекранний налагоджувач-симулятор асемблерних програм для однокристальних мікроконтролерів сімейства MCS-51.....	20
4.1. Загальні відомості про налагоджувач.....	20
4.2. Запуск налагоджувача та робота з ним.....	22
4.3. Робота з точками переривання.....	28
4.4. Функціональні клавіші та команди налагоджувача.....	29
Запитання для самоперевірки.....	32
Контрольне завдання для самоперевірки рівня опанування навичок роботи з симулятором.....	35
Індивідуальні завдання до самостійної роботи.....	36
Ресурси всесвітньої мережі Internet.....	40
Список рекомендованої літератури.....	41

ВСТУП

Мета самостійної та лабораторної робіт – подальше розширення і поглиблення знань з дисципліни “Мікропроцесорні пристрої”. До вивчення пропонується мікроконтролер (МК) 8051АН сімейства MCS-51 – один з найпоширеніших і найпопулярніших пристроїв сучасної мікроелектронної техніки.

Методичні вказівки, окрім навчального матеріалу, вміщують історичні відомості щодо створення та розвитку мікропроцесорів і мікроконтролерів. Ко-роткий огляд сучасного стану ринку однокристальних мікроконтролерів має переконати читача у необхідності і перспективності вивчення саме запропонованого сімейства МК. Далі наведені список літератури та адреси інформаційних сайтів виробників мікроконтролерів і програмного забезпечення у всесвітній мережі Internet.

Для практичного засвоєння навчального матеріалу і отримання початкових навичок програмування чіпа 8051АН слід скористатися досить простим налагоджувачем-симулятором FD51, який являє собою програмну модель мікроконтролера. Серед багатьох налагоджувачів перевага надана саме FD51 тому, що головне вікно програми зручно відображає внутрішні реєстри та елементи пам’яті мікроконтролера і дає уявлення про архітектуру. Крім цього, FD51 – дуже простий у користуванні. У наступних лабораторних роботах студенти вивчатимуть більш потужну програмну оболонку – інтегровану інструментальну систему IDE COMPASS/51/251.

1. ЗМІСТ САМОСТІЙНОЇ ТА ЛАБОРАТОРНОЇ РОБІТ

1.1. Самостійна робота

Самостійна робота полягає у вивченні теоретичного матеріалу, що викладений у даних методичних вказівках. Слід ознайомитися з призначенням і архітектурою мікроконтролера 8051АН фірми Intel, розглянути можливості налагоджувача FD51 та прийоми роботи з ним. Засвоївши матеріал, необхідно виконати контрольне завдання для самоперевірки рівня опанування навичок роботи з емулятором, відповіді на запитання для самоперевірки, наведені у кінці розділів 3 та 4 методичних вказівок, та виконати індивідуальне завдання до самостійної роботи.

1.2. Лабораторна робота

Виконується у комп’ютерних класах кафедри. До виконання допускаються студенти, які ознайомилися з теоретичним матеріалом (див. п 1.1) та попередньо підготували звіт. Кінцева мета – закріплення теоретичних знань та отримання практичних навичок роботи з симулятором-налагоджувачем FD51. Захист роботи – шляхом тестової перевірки на ПЕОМ та співбесіди з викладачем.

Програма виконання

- Після засвоєння матеріалу п.п. 2...3 методичних вказівок (МВ) відповіді на запитання для самоперевірки п. 3.
- Ознайомитися із призначення функціональних клавіш, команд і вікон налагоджувача-симулятора FD51 за допомогою персональної ЕОМ, отримати практичні навички та прийоми роботи з ним. Після засвоєння матеріалу п. 4 відповіді на запитання для самоперевірки рівня засвоєння знань.
- У режимі редагування асемблерних програм налагоджувача ввести програму прикладу і прослідкувати за зміною вмісту регістрів та пам'яті мікроконтролера при виконанні програми.
- Виконати індивідуальний варіант самостійної роботи (номер варіанту повинен збігатися з порядковим номером студента у журналі групи).
- Скласти підсумковий звіт та виконати завдання тестової перевірки.

Вказівки щодо складання звіту

Підсумковий звіт повинен містити:

- Назву, мету та програму роботи.
- Спрощену структурну схему мікроконтролера з його архітектурою.
- Відповіді на запитання самоперевірки.
- Команди налагоджувача-симулятора та їх формати.

Під час захисту роботи продемонструвати вміння працювати з налагоджувачем-симулятором, дію програми прикладу за контрольними вимогами викладача, які і визначають рівень опанування студентом навчального матеріалу.

2. ІСТОРІЯ СТВОРЕННЯ ТА РОЗВИТКУ

МІКРОПРОЦЕСОРІВ І МІКРОКОНТРОЛЕРІВ

Сучасні вирішення у галузі автоматизації, роботизації та електроприводу неможливо уявити без використання мікропроцесорних засобів та систем. Вагомий внесок у розвиток напівпровідникової мікросхемотехніки належить відомій американській фірмі Intel, заснованій у 1968 році. Це був час появи багатьох новітніх технологій, завдяки яким з'явилася можливість створювати мініатюрні напівпровідникові прилади – мікросхеми. Їх використання відкривало нові перспективи в усіх галузях техніки, зокрема і в автоматизації. Починалась ера цифрової машинної обробки інформації. Перший комп'ютер ENIAC, створений у 1946 році, важив близько 30 т і займав велике приміщення. У 1968 році в світі вже нараховувалося 30 тис. комп'ютерів. Це були переважно великі універсальні ЕОМ і “міні-комп'ютери” розміром з шафу. Неприємна

особливість цих ЕОМ – часті аварійні ситуації через перегрівання ламп та велику кількість роз'ємних з'єднань. Тому поява інтегральної електроніки зумовлена об'єктивними причинами.

Засновниками фірми Intel були талановиті вчені та винахідники Роберт Нойс, Гордон Мур та Ендрю Гроув. Саме Роберт Нойс у 1959 році винайшов інтегральну мікросхему. У середині 60-х років Нойс працював менеджером американської компанії Fairchild Semiconductor, що була відома своїми розробками у галузі електронних технологій. Гордон Мур очолював наукові дослідження та конструкторські розробки Fairchild Semiconductor, був одним з восьми засновників Fairchild. Енді Гроув, уродженець Угорщини, – спеціаліст з розробки технологічних процесів. Він прийшов у компанію Fairchild Semiconductor після того, як отримав в університеті Берклі ступінь доктора наук у галузі хімічних технологій.

Наприкінці 60-х багато талановитих інженерів звільнялися з Fairchild Semiconductor і створювали власні фірми. Роберт Нойс і Гордон Мур заснували Intel й стали першими її співробітниками. Згодом до них приєднався і Енді Гроув. Стартовий капітал (2,5 мільйона доларів) надав фірмі фінансист із Сан-Франциско Артур Рок.

Фірма Intel спеціалізувалася на виробництві напівпровідникових запам'ятовуючих пристроїв. Перший серійний виріб – мікросхема “3101” 64-розрядної Шоткі-біполярної статичної оперативної пам'яті. Особливе ж місце, яке посіла Intel у світі електроніки, пов'язане з іншими пристроями – мікропроцесорами. Саме вони стали технічною базою нинішньої комп'ютерної науково-технічної революції.

Поштовхом до створення мікропроцесорів виявився контракт між Intel і японською фірмою Busicom, що спеціалізувалася на виробництві калькуляторів. Busicom замовила Intel розробку дванадцяти спеціалізованих мікросхем, однак для виконання такого великого замовлення Intel не мала достатньо людських, фінансових та виробничих ресурсів. Тоді талановитий інженер Тед Хофф запропонував замість дванадцяти спеціалізованих мікросхем створити одну універсальну, що зможе їх замінити. Р. Нойс і Г. Мур зрозуміли витонченість запропонованого Т. Хоффом рішення. Ідея задовольнила і компанію Busicom, що фінансувала роботу. Таким чином, Intel почала розробку універсальної мікросхеми, яку можна запрограмувати на виконання тих чи інших команд. Уперше відпала необхідність в апаратній реалізації алгоритму роботи пристрою: всі операції з обробки числових даних відтепер виконувалися з перезаписувальною програмою, що обіцяло велику економію коштів та часу. Над реалізацією задуманого Т. Хоффом працювала група інженерів та конструкторів Intel, яку очолював Федеріко Феджин. Через 9 місяців напруженої праці з'явився перший в світі мікропроцесор 4004. Він налічував 2300 напівпровідникових транзисторів, але вільно вміщувався на долоні. За продуктивністю ж новий процесор не поступався комп'ютеру ENIAC, що займав 85 кубічних метрів і складався з 18000 вакуумних ламп. Тед Хофф розробив архітектуру першого процесора, Стен Мейзор – систему його команд, а Федеріко Феджин спроектував кристал процесора.

Зрозумівши переваги від використання мікропроцесорів, керівництво Intel пішло на переговори з компанією Busicom, внаслідок яких Intel придбала всі права на

процесор 4004 за 60 тисяч доларів (слід зазначити, що незабаром Busicom збанкрутувала). Після цього почалася широка рекламна компанія, метою якої було довести інженерному загалу великий потенціал програмованих мікропроцесорів у багатьох галузях – від керування дорожнім рухом до автоматизації складних виробничих процесів. Intel проводила семінари для інженерів, публікувала рекламні матеріали та довідкові посібники з використання мікропроцесорів. У деякі тижні фірма продавала більше довідкової документації, ніж самих мікропроцесорів. Через певний час вони знайшли дуже широке застосування.

Таким чином, мікросхема 4004 стала першим мікропроцесором. Приблизно через півроку про появу подібних пристроїв сповістили ще декілька фірм. Ці мікропроцесори, виконані за р-МОП технологією, були чотирирозрядними, тобто за одним заходом могли обробляти тільки 4 біти інформації. Довжина програм і набір команд – обмежені, перші процесори – без багатьох функцій, обов'язкових для сучасних мікропроцесорів. У 1972 році фірма Intel випустила процесор 8008, який успадкував усі основні риси 4004. Це – перший 8-розрядний процесор, який сьогодні відносять до процесорів першого покоління. Він уже мав акумулятор, шість регістрів загального призначення, показчик стеку, вісім регістрів адреси та спеціальні команди для введення/виведення даних, але цей процесор не став загальноживаним, зокрема у комерційних розробках.

Наприкінці 1973 року на фірмі Intel розробляється новий 8-розрядний мікропроцесор 8080. Його архітектура та система команд виявилися настільки вдалим, що й сьогодні він вважається класичним. Тому для вивчення принципів побудови мікропроцесорних засобів та систем найбільш придатний і нині – процесор та допоміжні мікросхеми саме серії 8080.

Широке застосування мікропроцесорів у техніці розпочалося з появою чіпа 8080, який належав до процесорів третього покоління, та він був не єдиним вдалим 8-розрядним процесором. Через півроку з'явився мікропроцесор 6800 американської фірми Motorola, який створив жорстку конкуренцію інтелівському процесору. Як і 8080, мікропроцесор 6800 був виконаний за n-МОП технологією, вимагав використання окремого тактового генератора, мав трьохшинну структуру з 16-розрядною шиною адреси, добре розвинену архітектуру і систему команд. Його головні переваги – потужніша, ніж у 8080, система переривань, та одна (а не три, як у 8080) напруга живлення. Принципи внутрішньої архітектури 6800 також значно відрізнялися від 8080. Насамперед відсутністю регістрів загального призначення, в яких, залежно від поставлених завдань, могла зберігатися як адресна інформація, так і числові дані. Замість них до складу внутрішніх пристроїв додався другий рівноцінний акумулятор для обробки даних та спеціалізовані 16-розрядні регістри, де зберігалася тільки адресна інформація. Дані для обробки вибиралися із зовнішньої пам'яті і туди ж поверталися після обробки. Команди роботи з пам'яттю були простішими й коротшими, але пересилання байта у пам'ять займало більше часу, ніж обмін даними між внутрішніми регістрами 8080. Архітектура жодного з двох згаданих процесорів не мала вагомих переваг, і кожен з них став родоначальником двох великих сімейств мікропроцесорів – Intel та Motorola, представники яких конкурують й досі.

Через рік після створення мікропроцесора 8080 кілька інженерів Intel перейшли в фірму Zilog і почали працювати над створенням нового процесора, спираючись на свої попередні розробки. В наслідок цього в 1977 році з'явився мікропроцесор Z80, який став кращим представником 8-розрядних процесорів. У порівнянні з 8080 він потребував однієї напруги живлення, мав більш потужну та гнучку систему переривань, втричі вищу максимальну тактову частоту, два акумулятори та подвійний набір регістрів загального призначення. Система команд Z80 вміщувала в себе всі 78 команд мікропроцесора 8080 і майже таку ж кількість додаткових команд, тому програми, створені для 8080, без будь-яких змін працюють і на Z80.

У 1978 році на фірмі Intel був виготовлений перший 16-розрядний мікропроцесор 8086, використаний компанією International Business Machines (IBM) для створення персональних комп'ютерів, а 16-розрядний чіп 68000 фірми Motorola застосований у відомих комп'ютерах Atari та Apple. Щодо “домашніх” комп'ютерів, то вони набули масового характеру з появою моделі ZX-Spectrum (на базі процесора Z80) англійської фірми Sinclair Research Ltd, засновником якої був талановитий інженер Клайв Сінклер. Ідея використовувати телевізор замість монітора і магнітофон для збереження програм і даних значно здешевила домашній комп'ютер і зробила його доступним середньому покупцеві.

Пізніше (середина 70-х) виникла ще одна тенденція у розвитку мікропроцесорів, що мала безпосереднє відношення до автоматизації та появи процесорів для вбудованих використань. Початок їй поклав процесор 8085 фірми Intel. Спочатку він був замислений як продовження чіпа 8080, а через деякий час з'явилися Z80 та новий мікропроцесор 6809 фірми Motorola. Обидва вони значно перевищували 8085 за продуктивністю, що спонукало Intel взятися за розробку першого 16-розрядного мікропроцесора 8086, та з розробкою периферійних мікросхем 8156 і 8755 процесор 8085 отримав нові перспективи. Перша мікросхема вміщувала статичний ОЗП обсягом 256 байт, два 8-розрядних, побітно налагоджуваних на введення/виведення порти і програмований таймер-лічильник. До складу другої входили три багаторозрядних порти введення/виведення та ПЗП ємністю 2 Кбайти з ультрафіолетовим затиранням. Об'єднавши відповідним чином виводи цих трьох мікросхем, розробники електронної апаратури отримали функціонально завершений модуль – мікроконтролер, який можна вбудовувати до будь-яких приладів: вольтметрів, частотометрів, різноманітних підсилювальних пристроїв та перетворювачів. Декілька фірм випустили економічні за живленням k-МОП версії цього сімейства. Це надало можливість створювати мікропроцесорні прилади з автономним живленням. Нарешті, наприкінці 70-х років Intel “об'єднала” ці три мікросхеми в один чіп і виробила однокристалну мікро-ЕОМ 8048, до складу якої ввійшли ОЗП та ПЗП, арифметико-логічний пристрій, вбудований тактовий генератор, таймер-лічильник, порти введення/виведення. Далі були створені подібні до 8048 мікроконтролери 8035 і 8748. Система команд однокристалних мікро-ЕОМ була значно слабшою, ніж у процесора 8085, об'єм ОЗП і ПЗП, кількість портів введення/виведення теж були меншими, ніж у вищезазначеного трикорпусного модуля, але все це розміщувалося в одному корпусі, що значно спрощувало розробку та виробниц-

тво нових пристроїв на базі однокристальних мікро-ЕОМ. Ідея створення універсальних апаратних засобів з програмним налагодженням на конкретні задачі, яка стала поштовхом до появи мікропроцесорів, отримала найвищий ступінь реалізації саме в однокристальних мікроконтролерах.

На початку 80-х років Intel випустила потужнішу однокристальну мікро-ЕОМ 8051, а згодом – і її модифікації 8031 та 8751. Ядро мікро-ЕОМ цієї серії – класичне для мікроконтролерів. Серед усіх мікроконтролерів сімейство MCS-51 – безсумнівний лідер за кількістю різновидів та компаній, де виготовляються його модифікації. На сьогодні існує понад 200 модифікацій мікроконтролерів MCS-51, які випускають майже 20 провідних фірм-виробників електронних компонентів (Intel, Atmel, Siemens, Philips, Dallas Semiconductor, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems), а також підприємства України, Росії, Білорусі). Отримали поширення і мікроконтролери оригінальної архітектури фірм Motorola, Zilog, Analog Devices, Microchip, Scenix, Holtec.

Створення мікропроцесора визнано одним із визначних досягнень XX сторіччя. Тільки у 1995 році в світі було продано більше 200 мільйонів мікропроцесорів і понад 3 мільярдів мікроконтролерів. За даними журналу “Світ комп’ютерної автоматизації”, середній американець протягом дня близько 300 разів (!) має справу з мікроконтролерами, вбудованими всюди – від пральних машин, ліфтів і телефонів до світлофорів, автомобілів та промислових верстатів.

Журнал “Огляд стану справ у напівпровідниковій промисловості і торгівлі” (“Semiconductor Industry and Business Survey”) вважає: якби автомобілебудування та авіаційна промисловість розвивалися такими ж темпами, як виробництво напівпровідників протягом останніх 30 років, то автомобіль “Ролс-Ройс” коштував би 2 долари 75 центів і, використавши лише один літр бензину, міг би проїхати майже півтори тисячі кілометрів, а літак “Боїнг 767” коштував би 500 доларів і міг би облетіти навколо земної кулі за 20 хвилин, витративши лише каністру гасу. У 1996 році імена творців мікропроцесора доктора Теда Хоффа, доктора Федеріко Феджина та Стена Мейзора були занесені до Національної зали слави винахідників США (м. Ейкрон, Огайо) і стали поряд із іменами Томаса Едісона, братів Райт та Олександра Белла.

Незважаючи на інтенсивний розвиток та появу нових 16- та 32-розрядних мікроконтролерів і мікропроцесорів для вбудованих використань, найбільша частина світового мікропроцесорного ринку й досі залишається за 8-розрядними пристроями. За даними Semic Research Corp., Phoenix, середньорічний загальний світовий обсяг продаж мікроконтролерів усіх типів становив 11,4 млрд доларів, при цьому 5,56 млрд (або 48,6%) припало на частку 8-розрядних кристалів. Це у 2,5 раза більше обсягу продаж найближчих конкурентів: 16-розрядних мікроконтролерів (2,1 млрд дол.) та сигнальних процесорів (2,4 млрд дол.). За всіма прогнозами аналітичних компаній, лідируюче положення 8-розрядних мікроконтролерів на світовому ринку зберігатиметься і надалі.

3. СТРУКТУРНА ОРГАНІЗАЦІЯ МІКРОКОНТРОЛЕРІВ СІМЕЙСТВА MCS-51

3.1. Загальні відомості про архітектуру мікроконтролера

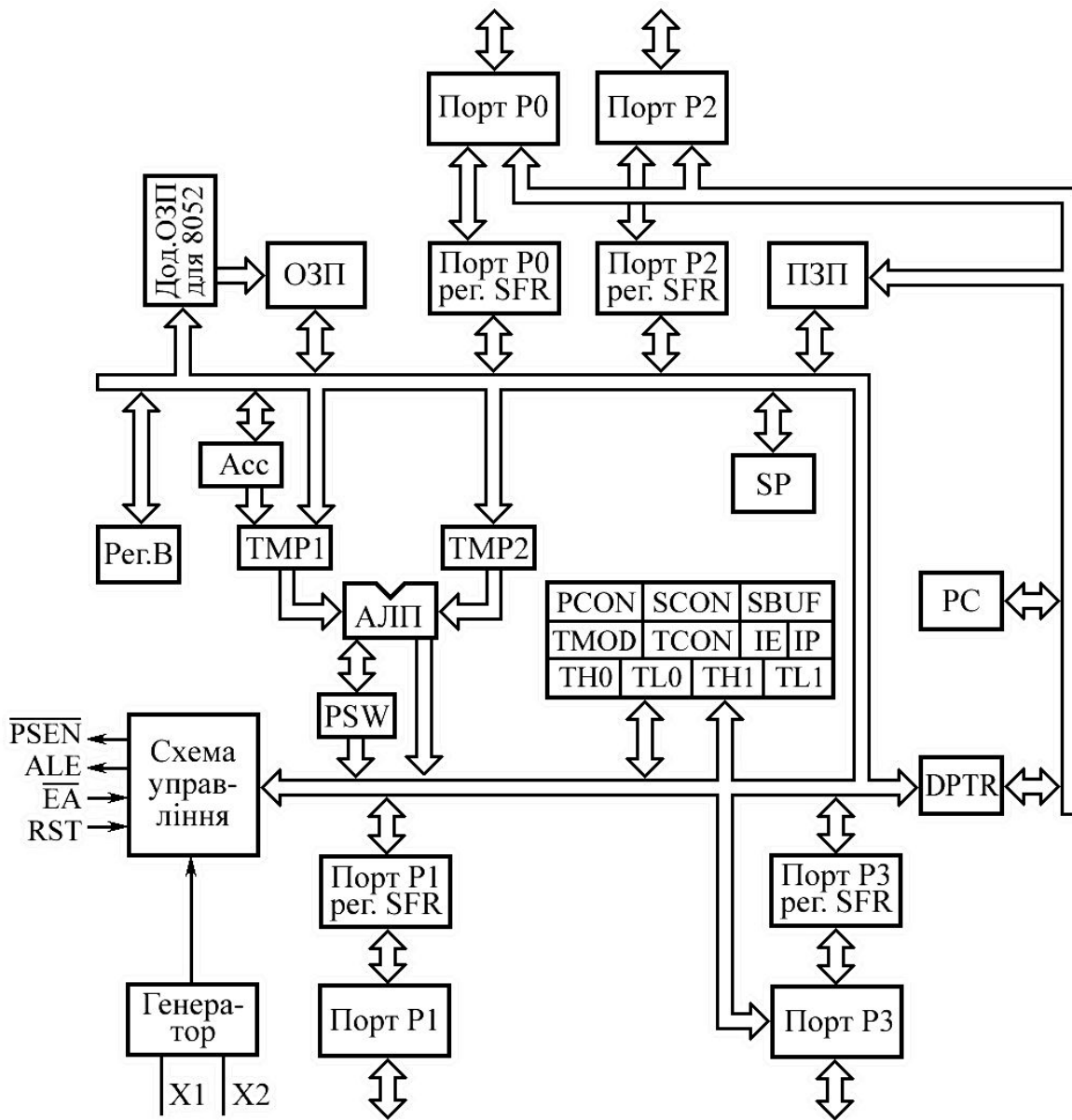
Однокристальний мікроконтролер для вбудованих використань (embedded microcontroller), який раніше інколи називали однокристальною мікро-ЕОМ, являє собою виготовлену на одному кристалі мікропроцесорну систему, зорієнтовану на реалізацію алгоритмів цифрового керування різноманітними об'єктами та процесами. У складі мікроконтролера, як правило, – центральний процесор, внутрішня постійна та оперативна пам'ять, паралельні та послідовні порти введення/виведення даних, набір периферійних пристроїв: таймерів, контролерів переривань, аналого-цифрових перетворювачів, широтно-імпульсних модуляторів, модулів цифрової обробки сигналів у реальному часі. Таким чином, на базі однієї мікросхеми мікроконтролера з використанням мінімальної кількості зовнішніх елементів можна побудувати багатофункціональну універсальну цифрову систему.

За сукупністю функціональних можливостей та технічних характеристик мікроконтролери зорієнтовані перш за все на вирішення завдань управління різноманітними приладами та пристроями. Вони використовуються також як компоненти у системах управління електроприводами, інформаційно-вимірювальних та контрольно-діагностичних системах, побутовій техніці, автомобільному, залізничному та повітряному транспорті, військовій техніці тощо.

Найбільш суттєва відмінність архітектури однокристальних мікроконтролерів від традиційної архітектури мікропроцесорів у тому, що внутрішня пам'ять програм (ВПП) та внутрішня пам'ять даних (ВПД) – це принципово різні пристрої. Якщо мікропроцесор КР580ВМ80 здатний виконувати коди програм, які можуть перебувати як в постійній, так і в оперативній пам'яті, то мікроконтролер виконує програму, яка тільки в пам'яті програм. Коли мікропроцесор використовує одні й ті ж сигнали \overline{RD} і \overline{WR} при звертанні до мікросхем як постійної, так і оперативної пам'яті, то однокристальний мікроконтролер встановлює різні сигнали при звертанні до зовнішніх пам'яті програм та пам'яті даних.

На *рис. 3.1* наведена внутрішня структура мікроконтролера 8051АН – вихідна для сімейства MCS-51, що вміщує такий набір функціональних модулів:

- 8-розрядний арифметико-логічний пристрій (АЛП) з апаратною реалізацією операцій множення/ділення;
- внутрішній оперативний запам'ятовуючий пристрій (ОЗП, пам'ять даних) об'ємом 128 байтів;
- внутрішній постійний запам'ятовуючий пристрій (ПЗП, пам'ять програм) об'ємом 4 Кбайти;
- чотири двоспрямованих побітно налагоджуваних 8-розрядних порти введення/виведення;
- два 16-розрядних таймери-лічильники;



Порт P0...Порт P3 – порти мікроконтролера; **Per. SFR** – регістри портів мікроконтролера; **ОЗП** – оперативний запам'ятовуючий пристрій (пам'ять даних); **ПЗП** – постійний запам'ятовуючий пристрій (пам'ять програм); **Дод. ОЗП** – додаткова пам'ять даних (для мікроконтролерів i8052; **Acc** – акумулятор; **Per. B** – регістр-розширювач акумулятора (регістр B); **TMP1**, **TMP2** – регістри тимчасового зберігання даних; **SP** – регістр-показчик стеку; **PCON**, **SCON**, **SBUF**, **TMOD**, **TCON**, **IE**, **IP**, **TH0**, **TL0**, **TH1**, **TL1** – службові регістри спеціальних функцій внутрішніх пристроїв мікроконтролера; **PC** – програмний лічильник команд; **PSW** – регістр слова стану програми; **DPTR** – регістр-показчик даних

Рис.3.1. Внутрішня структура однокристального мікроконтролера 8051AH сімейства MCS-51

- дуплексний послідовний порт для зв'язку з іншими контролерами чи мікро-ЕОМ;

- вбудований тактовий генератор.

Мікроконтролер 8051АН має особливості, обумовлені його апаратною реалізацією:

- адресація 64 Кбайтів зовнішньої пам'яті програм та 64 Кбайтів зовнішньої пам'яті даних;

- можливість виконання лініями порта 3 альтернативних функцій;

- можливість виконання операцій над окремими бітами;

- 2 лінії запитів на переривання від зовнішніх пристроїв;

- неможливість зупинки роботи мікроконтролера після подання напруги живлення і відсутність команди зупинки для мікросхем, створених з використанням схемотехніки динамічних елементів;

- наявність двох режимів зниженого енергоспоживання;

- наявність біта захисту від несанкціонованого копіювання програми із внутрішнього ПЗП.

Слід мати на увазі, що технічні дані та архітектура мікроконтролерів типу 8051АН різних фірм-виробників електронних компонентів можуть відрізнятися.

На *рис. 3.2* зображено розташування та призначення виводів мікроконтролера 8051АН:

U_{SS} – потенціал загального проводу (“землі”);

U_{CC} – основна напруга живлення +5 В;

X1, X2 – виводи для під'єднання кварцового резонатора;

RST – вхід загального скидання мікроконтролера;

\overline{PSEN} – дозвіл зовнішньої пам'яті програм; видається тільки при зверненні до зовнішнього ПЗП;

\overline{ALE} – строб адреси зовнішньої пам'яті;

\overline{EA} – дозвіл внутрішньої пам'яті програм; рівень 0 на вході змушує виконувати програму із зовнішнього ПЗП, ігноруючи внутрішню пам'ять програм;

P1 – 8-бітовий квазідвоспрямований порт введення/виведення; кожен біт порту може бути незалежно налагоджений на введення/виведення інформації;

P2 – 8-бітовий квазідвоспрямований порт, аналогічний P1; крім того, виводи цього порту використовують для видачі старшого байта адреси при зверненні до зовнішньої пам'яті програм або даних за 16-бітової адресації. Виводи порту застосовуються для введення старших біт адреси при програмуванні внутрішнього ПЗП (для мікросхем, що програмуються користувачем);

P3 – 8-бітовий квазідвоспрямований порт, аналогічний до P1; крім того, виводи цього порту можуть виконувати ряд альтернативних функцій, описаних нижче;

P0 – 8-бітовий двоспрямований порт введення/виведення інформації; при роботі із зовнішніми ОЗП та ПЗП лініями порту послідовно видається молодший байт адреси зовнішньої пам’яті, після чого дані передаються чи приймаються.

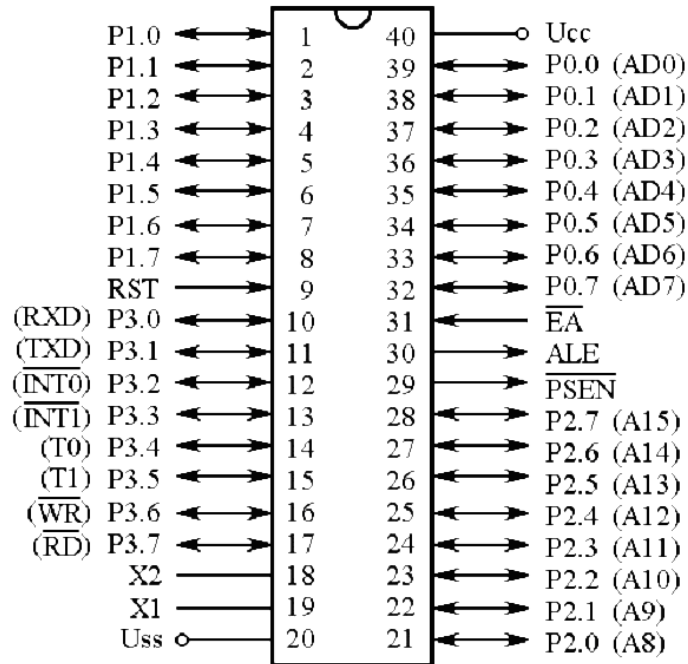


Рис. 3.2. Розташування виводів мікроконтролера 8051AH

Розширені версії мікроконтролерів сімейства MCS-51 можуть мати наступні периферійні пристрої:

- 32-розрядний математичний співпроцесор;
- додатковий 16-розрядний регістр DPTR;
- порт сканування клавіатури;
- контролер мультиплексного рідинно-кристалічного індикатора;
- додатковий багаторежимний таймер-лічильник PCA;
- сторожевий таймер (Watch Dog Timer, WDT);
- аналоговий компаратор;
- багатоканальний аналого-цифровий перетворювач (АЦП);
- прецизійний 24-розрядний АЦП;
- двоканальний цифро-аналоговий перетворювач (ЦАП);
- додатковий дуплексний асинхронний приймач-передавач (UART);
- додаткові послідовні інтерфейси (SPI, I2C, SMBus, CAN);
- “комп’ютерні” інтерфейси (USB, eISA);
- енергонезалежна пам’ять даних on-chip EEPROM (до 128 Кбайт);
- конфігурована логічна матриця CSL (до 2048 елементів);
- декодер програвача MP-3.

Сучасні мікроконтролери сімейства MCS-51 мають повністю статичну архітектуру і демонструють рекордну для 8-розрядних пристроїв швидкодію до 100 мільйонів операцій за секунду (MIPS).

Широкому розповсюдженню мікроконтролерів сімейства MCS-51 сприяли такі переваги:

- архітектура, яка фактично стала промисловим стандартом;
- велика широта та різноманітність можливостей;
- наявність високопродуктивних та розширених версій процесорів;
- значна кількість вільно доступних програмних та апаратних розробок;
- легкість апаратного програмування, у тому числі і внутрішньосхемного;
- дешевина базових чіпів сімейства;
- наявність спеціалізованих версій контролерів;
- наявність версій контролерів зі зниженим рівнем електромагнітних завад;
- доступність спеціальної літератури та інформації в Internet;
- підтримка архітектури провідними навчальними закладами світу.

Сьогодні мікроконтролери сімейства MCS-51 виробляють більше 20 фірм, серед яких найбільш відомі: Analog Devices; Atmel Corporation; Burr-Brown Products, підрозділ Texas Instruments; Cypress Semiconductor; Cygnal Integrated Products, Inc.; Dallas Semiconductor; Infineon Technologies AG – (раніше Siemens Semiconductors AG, підрозділ Siemens AG); Integrated Silicon Solution, Inc.; Philips Semiconductors; ST Microelectronics; Texas Instruments, Inc.; Triscend Corporation; Winbond Electronics Corp.; Temic Semiconductor (підрозділ Atmel Corp.), OKI Semiconductor, LG Electronics, Cybernetic Micro Systems, Matra MHS, Silicon Systems (SSI), Advanced Micro Devices (AMD), та інші.

3.2. Організація ОЗП, ПЗП та реєстрів мікроконтролера

Пам'ять програм та даних мікроконтролера – це самостійні пристрої, що адресуються різними командами та керуючими сигналами. Об'єм пам'яті програм, що розташована на кристалі мікроконтролера 8051АН, дорівнює 4 Кбайти. Деякі мікроконтролери сімейства MCS-51 мають до 32 Кбайтів внутрішньої пам'яті програм. При зверненні до зовнішньої пам'яті програм мікроконтролер завжди використовує 16-розрядну адресу, що забезпечує доступ до 64 Кбайтів ПЗП. Мікроконтролер звертається до програмної пам'яті при читанні коду операції і операндів (за допомогою лічильника команд РС), а також при виконанні команд перенесення байта із пам'яті програм до акумулятора. При виконанні команд перенесення даних адресація елемента пам'яті програм, із якого будуть зчитані дані, може здійснюватись з використанням як програмного лічильника РС, так і спеціального дво-байтового реєстра-показчика даних DPTR. Об'єм розташованої на кристалі 8051АН пам'яті даних – 128 байтів. Інші модифікації мікроконтролерів сімейства

MCS-51 можуть мати 256 байтів внутрішнього ОЗП. Перші 32 байти внутрішньої пам'яті даних згруповані до чотирьох банків регістрів загального призначення, які позначаються як *банк 0... банк 3*. Кожен з них складається з восьми регістрів R0...R7. У кожен момент часу програмі доступний тільки один банк регістрів, номер якого вміщують третій та четвертий біти слова стану програми PSW (*рис. 3.3*).

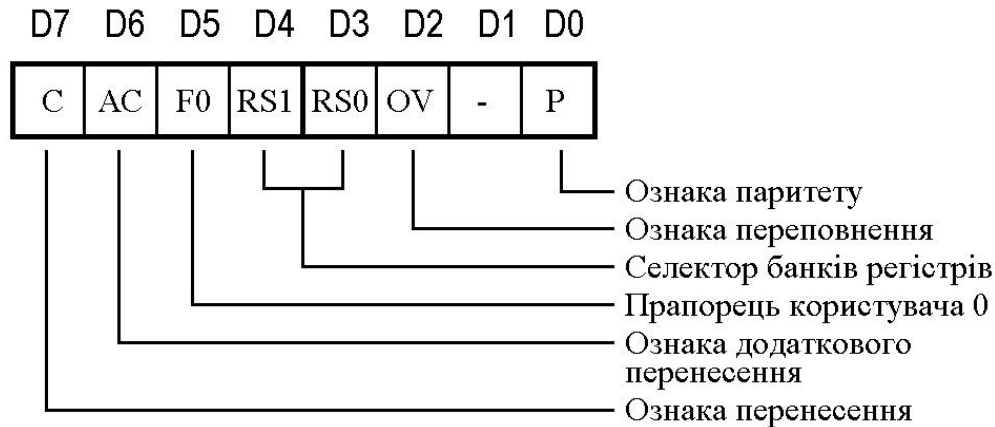


Рис. 3.3. Прапорці слова стану програми PSW

Решта адресного простору може бути сконфігурована користувачем за власним розсудом: у ньому можна розташувати стек, системні дані та дані користувача. Звернення до елементів пам'яті даних можливе двома способами: перший – пряма адресація елементів пам'яті (адреса елемента є операндом відповідної команди); другий – посередня (непряма) адресація за допомогою регістрів R0 або R1: перед виконанням відповідної команди до одного з них повинна бути занесена адреса елемента, куди необхідно звернутися.

До адресного простору пам'яті даних примикає адресний простір регістрів спеціальних функцій (SFR). Адреси, за якими розташовані ці регістри, наведені в *табл. 3.1*. Регістри спеціальних функцій займають тільки частину 128-байтового адресного простору. Елементи пам'яті з адресами 80H...0FFH, які не зайняті регістрами, фізично відсутні на кристалі мікроконтролера 8051AH і можуть бути використані в інших модифікаціях сімейства MCS-51.

При зверненні до зовнішньої пам'яті даних мікроконтролер може адресувати 64 Кбайти ОЗП. Адресація цих елементів зовнішнього ОЗП здійснюється посередньо регістрами R0 і R1 або двобайтовим регістром DPTR. Звернення до зовнішньої пам'яті даних виконується за допомогою сигналів запису або читання \overline{WR} та \overline{RD} (*табл. 3.2*).

Акумулятор є джерелом операнда і місцем фіксації результату при виконанні ряду операцій. Тільки з використанням акумулятора можна здійснити операції зсуву, перевірки на нуль тощо. Але на відміну від більшості мікропроцесорів система команд мікроконтролерів сімейства MCS-51 вміщує велику кількість команд пересилки, переходів і логічних операцій, що здійснюються без використання

Таблиця 3.1

Назва та адреси спеціальних регістрів мікроконтролера 8051АН

Символ	Адреса	Назва
ACC *	0E0H	Акумулятор
B *	0F0H	Регістр-розширювач акумулятора
PSW *	0D0H	Слово стану програми
P0 *	080H	Порт 0 (SFR P0)
P1 *	090H	Порт 1 (SFR P1)
P2 *	0A0H	Порт 2 (SFR P2)
P3 *	0B0H	Порт 3 (SFR P3)
SP	081H	Регістр-показчик стеку
DPH	083H	Старший байт регістра-показчика даних DPTR
DPL	082H	Молодший байт регістра-показчика даних DPTR
TH0	08CH	Старший байт таймера 0
TL0	08AH	Молодший байт таймера 0
TH1	08DH	Старший байт таймера 1
TL1	08BH	Молодший байт таймера 1
TMOD	089H	Регістр режимів таймерів-лічильників
TCON *	088H	Регістр управління/ статусу таймера
IP *	0B8H	Регістр пріоритетів переривань
IE *	0A8H	Регістр маски переривань
PCON	087H	Регістр управління електроспоживанням
SCON *	098H	Регістр управління приймачем-передавачем
SBUF	099H	Буфер приймача-передавача

Примітка: регістри, імена яких позначені знаком (*), допускають адресацію окремих бітів при виконанні команд із групи команд операцій над бітами.

Таблиця 3.2

Альтернативні функції виводів портів

Вивід порту	Альтернативна функція
P3.0	RXD – вхід послідовного порту TXD – вихід послідовного порту
P3.1	
P3.2	$\overline{\text{INT0}}$ – зовнішнє переривання 0
P3.3	$\overline{\text{INT1}}$ – зовнішнє переривання 1
P3.4	T0 – вхід таймера-лічильника 0
P3.5	T1 – вхід таймера-лічильника 1
P3.6	$\overline{\text{WR}}$ – строб запису до зовнішньої пам'яті даних
P3.7	$\overline{\text{RD}}$ – строб читання із зовнішньої пам'яті даних

акумулятора. Це забезпечує гнучкість програмування. При виконанні ряду команд в арифметико-логічному пристрої (АЛП) формуються ознаки операцій – прапорці, які фіксуються в регістрі PSW. Перелік прапорців, їх символічні імена та умови формування – у *табл. 3.3*.

Таблиця 3.3

Призначення прапорців слова стану програми

Символ	Позиція	Ім'я та призначення			
P	PSW.0	Прапорець паритету. Встановлюється в 1 за умови непарної кількості одиничних бітів акумулятора та в 0 при парній їх кількості			
-	PSW.1	Зарезервований біт			
OV	PSW.2	Прапорець переповнення. Встановлюється і скидається при виконанні арифметичних дій			
RS0	PSW.3	Біти вибору банку регістрів. Можуть бути встановлені або скинуті програмним шляхом			
RS1	PSW.4	RS0	RS1	Банк	Адреси регістрів банку
		0	0	0	00H – 07H
		1	0	1	08H – 0FH
		0	1	2	10H – 17H
		1	1	3	18H – 1FH
F0	PSW.5	Прапорець користувача. Може бути встановлений, скинутий і перевірений програмою користувача			
AC	PSW.6	Прапорець допоміжного перенесення. Встановлюється та скидається при виконанні команд додавання і віднімання при перенесенні чи запозиченні із старшої тетради			
C	PSW.7	Прапорець перенесення. Встановлюється та скидається як апаратним, так і програмним шляхом			

Регістр-показчик стеку SP мікроконтролера 8051AH – 8-бітовий. Він може адресувати будь-яку область внутрішньої пам'яті даних. У мікроконтролерів сімейства MCS-51 стек “росте вгору”, тобто перед виконанням команди PUSH чи CALL вміст регістра SP інкрементується (збільшується на 1), після чого інформація записується до стеку. Відповідно при вибиранні інформації із стеку регістр SP декрементується після вибирання байта даних.

Під час ініціалізації мікроконтролера після скидання або при вмиканні напруги живлення до SP заноситься код 07H. Таким чином, перший елемент даних стеку буде розміщений за адресою 08H.

Регістр-показчик даних DPTR частіше всього використовують для фіксації 16-бітової адреси в операціях звернення до зовнішньої пам'яті програм і даних, а з точки зору програміста – у вигляді одного 16-бітового регістра та двох незалежних однобайтових регістрів DPH і DPL.

Дві регістрові пари TH0 та TL0 і TH1 та TL1 являють собою регістри, що забезпечують незалежне функціонування двох програмно-керованих 16-бітових таймерів-лічильників. Режими роботи останніх задаються за допомогою регістра TMOD, а керування здійснюється регістром TCON. Для керування режимами енергоживання мікросхеми вживається регістр PCON. Регістри IP і IE керують роботою системи переривань мікроконтролера, а SBUF і SCON – приймача-передавача послідовного порту.

Запитання для самоперевірки

- Який мікроконтролер є базовим чіпом сімейства MCS-51 фірми Intel?
- Які особливості однокристальних мікроконтролерів забезпечили їх широке застосування?
- Назвіть головні риси мікроконтролерів, завдяки яким їх класифікують як окрему від мікропроцесорів групу цифрових пристроїв.
- Скільки портів має мікроконтролер 8051АН ?
- Чи допускають порти мікроконтролера 8051АН побітну модифікацію окремих розрядів?
- Скільки входів запитів на переривання у мікроконтролера 8051АН ?
- Чи є у складі мікроконтролера 8051АН апаратні засоби для передачі інформації у послідовному форматі?
- Які мікросхеми необхідні додатково для мікроконтролера 8051АН при мінімальній конфігурації системи?
- Чи передбачена можливість роботи мікроконтролера 8051АН із зменшеним енергоживленням?
- Скільки джерел напруги необхідно для живлення мікроконтролера в режимі виконання програми?
- Скільки банків регістрів загального призначення має мікроконтролер 8051АН ?
- Яке значення заноситься до регістра-показчика стеку під час початкової ініціалізації мікроконтролера?
- За допомогою якого регістра задаються режими роботи таймерами?
- Який об'єм пам'яті програм може бути доступним для мікроконтролера?
- Який об'єм пам'яті даних може бути доступним для мікроконтролера?
- Який діапазон тактових частот мікроконтролера?
- Чи є можливість збільшити кількість запитів на переривання?

Bank 0	Bank 1	Bank 2	Bank 3	Special Function Registers	
R0=00->00	R0=00->00	R0=00->00	R0=00->00	TH0= 00	TL0= 00
R1=00->00	R1=00->00	R1=00->00	R1=00->00	TH1= 00	TL1= 00
R2=00	R2=00	R2=00	R2=00	P0= FF	P1= FF
R3=00	R3=00	R3=00	R3=00	P2= FF	P3= FF
R4=00	R4=00	R4=00	R4=00	DPH= 00	DPL= 00
R5=00	R5=00	R5=00	R5=00	SP= 07	IP= 10100000
R6=00	R6=00	R6=00	R6=00	TMOD=00000000	IE= 01000000
R7=00	R7=00	R7=00	R7=00	TCON=00000000	SCON=00000000
A=00	B=00		PC=0000	SBUF=00	PSW= 02
			INT RAM	P S W	
			0000 00 00 00 00 00 00	C AC F0 S1 S0 OV ** P	
0000 00	NOP		0006 00 00 00 00 00 00	0 0 0 0 0 0 1 0	
0001 00	NOP		000C 00 00 00 00 00 00	PGM ROM	
0002 00	NOP		0012 00 00 00 00 00 00	0000 00 00 00 00	Stack
0003 00	NOP		0018 00 00 00 00 00 00	0004 00 00 00 00	07 00
0004 00	NOP		001E 00 00 00 00 00 00	0008 00 00 00 00	06 00
0005 00	NOP		0024 00 00 00 00 00 00	000C 00 00 00 00	05 00
0006 00	NOP		002A 00 00 00 00 00 00	0010 00 00 00 00	04 00
0007 00	NOP		0030 00 00 00 00 00 00	0014 00 00 00 00	03 00
0008 00	NOP		0036 00 00 00 00 00 00	0018 00 00 00 00	02 00
0009 00	NOP			S ms mcs	
000A 00	NOP		CMD >_	000 000 000	01 00
000B 00	NOP				00 00
1 Step 2 Proc 3Reg:DEC4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓					

Bank 0	Bank 1	Bank 2	Bank 3	Special Function Registers
R0=00->00	RA=00->00	R0=00->00	R0=00->00	TH0= 00
R1=00->00		R1=00->00	R1=00->00	TH1= 0
R2=00	РЕГІСТРИ	R2=00	R2=00	P0=
R3=00	ЗАГАЛЬНОГО	R3=00	R3=00	P2=
R4=00	ПРИЗНАЧЕННЯ	R4=00	R4=00	DP1=
R5=00		R5=00	R5=00	SP=
R6=00		R6=00	R6=00	TMOD=0
R7=00		R7=00	R7=00	TCON=00000000
A=00	B=00	PC=0000		SBUF=00

INT RAM	Stack
0000 00 00 00 00 00 00 00	С ПРАПОРЦІ P 0
0001 00 NOP	
0002 00 NOP	
0003 00	
0004 00	
0005 00	
0006 00	
0007 00	
0008 00	
0009 00	
000A 00	
000B 00	
000C 00	
000D 00	
000E 00	
000F 00	
0010 00	
0011 00	
0012 00	
0013 00	
0014 00	
0015 00	
0016 00	
0017 00	
0018 00	
0019 00	
001A 00	
001B 00	
001C 00	
001D 00	
001E 00	
001F 00	
0020 00	
0021 00	
0022 00	
0023 00	
0024 00	
0025 00	
0026 00	
0027 00	
0028 00	
0029 00	
002A 00	
002B 00	
002C 00	
002D 00	
002E 00	
002F 00	
0030 00	
0031 00	
0032 00	
0033 00	
0034 00	
0035 00	
0036 00	
CMD РЯДОК КОМАНД	ЛЧИЛЬНИК ЧАСУ 00

1 Step	РЯДОК МЕНЮ	5SetBrk	меню: BIN/ I/E	Pgm ↑	↓	Pgm ↓
--------	------------	---------	----------------	-------	---	-------

6

Рис. 4.1. Головне вікно налагоджувача FD-51

4. ПОВНОЕКРАННИЙ НАЛАГОДЖУВАЧ-СИМУЛЯТОР АСЕМБЛЕРНИХ ПРОГРАМ ДЛЯ ОДНОКРИСТАЛЬНИХ МІКРОКОНТРОЛЕРІВ СІМЕЙСТВА MCS-51

4.1. Загальні відомості про налагоджувач

Повноекранний налагоджувач-симулятор FD51 для програм, написаних мовою АСЕМБЛЕРА однокристальних мікроконтролерів сімейства MCS-51, призначений для перегляду і налагодження цих програм. Налагоджувач – це програмна модель мікроконтролера 8051АН, яка працює на персональних ЕОМ типу ІВМ РС ХТ/АТ і сумісних з ними.

Налагоджувач дає можливість:

- набирати текст програми мовою АСЕМБЛЕРА у мнемонічних кодах машинних команд;
- завантажувати для налагодження файли з машинними двійковими кодами асемблерних програм;
- переглядати на екрані текст завантаженої програми, включаючи адреси і коди команд, вміст усіх регістрів мікроконтролера, елементів пам'яті даних та пам'яті програм, вміст зовнішньої оперативної та стекової пам'яті;
- виконувати завантажену програму по кроках з переглядом результатів після кожного кроку і у безперервному режимі із зупинками у точках переривання;
- вносити зміни до завантаженої програми у мнемонічних кодах мови АСЕМБЛЕРА або у машинних кодах;
- змінювати вміст регістрів, прапорців слова стану програми і елементів пам'яті у командному режимі та режимі повноекранного редагування;
- зберігати вміст будь-якої області пам'яті у файлі на дисковому носії;
- виводити на друк або дискові носії текст програми і дамп пам'яті;
- завантажувати пам'ять з дискового файлу;
- отримувати трасування програми;
- визначати час виконання завантаженої програми та її частин за допомогою вбудованого лічильника.

4.2. Запуск налагоджувача та робота з ним

Для завантаження програми налагоджувача-симулятора необхідно запустити файл fd51.exe (про місцезнаходження файлу можна дізнатися у викладача). Відразу ж після запуску налагоджувач-симулятор FD51 готовий до роботи (за малого

розміру робочого вікна – меншого екрану дисплея – слід одночасно натиснути клавіші [*Alt*] і [*Enter*]), а на екран ЕОМ виводиться головне робоче вікно програми (рис. 4.1). Воно висвітлює вміст чотирьох банків регістрів загально-го призначення (Bank0...Bank3); блоку регістрів спеціальних функцій (Special Function Registers); пам'яті даних (INT RAM) та програм (PGM ROM); байта стану програми (PSW); стекової пам'яті (Stack) та лічильника часу (s-ms-mcs). У лівій верхній частині головного вікна налагоджувача можна побачити вміст регістрів непрямої адреси R0, R1 і елементів пам'яті, що адресуються цими регістрами:

Bank 0	Bank 1
R0=00->00	R0=00->00
R1=00->00	R1=00->00
R2=00	R2=00

У лівій нижній частині головного вікна налагоджувача знаходиться вікно редагування тексту програм. Процесом налагоджування можна керувати командами меню та налагоджувача, що вводяться у командному рядку (нижня частина головного вікна).

Команди вводяться користувачем за допомогою алфавітно-цифрових клавіш клавіатури персональної ЕОМ. При цьому можна користуватися клавішами вставки [*Insert*]; видалення [*Del*], [*BackSpace*]; позиціонування курсору [←], [→]; відміни [*Esc*]. Необхідно мати на увазі, що після початку введення команди і до натиснення клавіші [*Enter*] функціональні клавіші [*F1...F10*] недоступні, тобто рядок команд меню не працює. Якщо команда, що вводиться, помилкова, видається повідомлення про це і звуковий сигнал (призначення функціональних клавіш та команди налагоджувача описані нижче).

Для переведення налагоджувача у режим повноекранного редагування необхідно натиснути клавішу [*Enter*] без введення команди. При цьому курсор набуває вигляду мерехтливого прямокутника ■. Тепер можна переміщувати курсор по екрану завдяки клавішам керування курсором і змінювати вміст регістрів, пам'яті і прапорців введенням нових значень цифровими клавішами. Можна змінювати також початкову адресу поточної інструкції асемблерного редагованого тексту і початкові адреси вікон пам'яті (у перших рядках вікон). Повноекранне редагування можна виконувати при шістнадцятковому, десятковому та двійковому поданні інформації на екран. Під час редагування залишаються доступними всі команди меню, що вводяться функціональними клавішами [*F1...F10*]. Для швидкого переміщення курсору по екрану користуватися клавішами [*Tab*] і [*Shift*]+[*Tab*]. Для того, щоб повернутися до командного рядка, необхідно повторно натиснути клавішу [*Enter*]. При цьому режим повноекранного редагування вимикається.

Для переходу до режиму АСЕМБЛЕРА (для введення команд налагоджуваної програми у мнемонічних кодах) спочатку необхідно увімкнути режим повноекранного редагування одноразовим натисненням клавіші [*Enter*], після чого помістити курсор в першу (ліву) позицію поля поточної інструкції (команди) і за допомогою клавіш встановити початкову адресу програми у вигляді чотирьох цифр шістнадцяткового коду. Оскільки до цього моменту програма користувача ще не завантажена, то АСЕМБЛЕР після введеного коду адреси (наприклад, 0000) висвітлює ще дві цифри 00 шістнадцяткового коду команди NOP і, власне, її мнемоніку. При цьому весь цей командний рядок затемнений.

Для того щоб ввести за вибраною адресою мнемоніку нової команди спочатку слід перевести мерехтливий курсор в першу (ліву) позицію коду замінюваної команди (у випадку, коли треба змінити команду NOP, курсор повинен бути розміщений в позиції N). При цьому всі символи замінюваної мнемоніки команди будуть яскраво висвітлені:

```

0000 00  NOP
0001 00  NOP
0002 00  NOP

```

Тепер наберіть мнемоніку, наприклад, MOV A,B (команди АСЕМБЛЕРА можуть набиратися як великими, так і малими літерами):

```

0000 00  MOV A,B
0001 00  NOP
0002 00  NOP

```

і натисніть [*Enter*]. Якщо мнемоніка правильна, то відповідні коди команди (E5 – код операції та F0 – адреса регістра B) послідовно заносяться до пам'яті програм (PGM ROM) за адресами 0000H та 0001H, а курсор вікна встановлюється на наступну адресу байта, куди буде заноситися код операції наступної команди, а саме 0002H:

```

0000 E5F0  MOV  A,B
0002 00  NOP
0003 00  NOP
0004 00  NOP

```


Якщо мнемонічний код команди набрано невірно, у командному рядку висвітлюється повідомлення про помилку у вигляді **Invalid mnemonic**:

```

0036 00 00 00 00 00 00 | 0014 00 00 00 00 00 | 03 00
0036 00 00 00 00 00 00 | 0018 00 00 00 00 00 | 02 00
Invalid mnemonic | s ms mcs | 01 00
CMD > | 000 000 000 | 00 00
4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓

```

Якщо ж набрано помилкові операнди команди, то видається повідомлення *Invalid operand (s)*:

```

0036 00 00 00 00 00 00 | 0014 00 00 00 00 00 | 03 00
0036 00 00 00 00 00 00 | 0018 00 00 00 00 00 | 02 00
Invalid operand(s) | s ms mcs | 01 00
CMD > | 000 000 000 | 00 00
4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓

```

Помилкові символи можна видалити розміщенням на них курсору та натисканням клавіші «Пропуск». Можливо і перемістити курсор лівіше першої букви введеної команди (і тим самим замінити цю команду на попередню) із зняттям повідомлення про помилку.

Помилково набрана адреса переходу супроводжується звуковим сигналом та попередженням **Address cannot be reached**:

```

0036 00 00 00 00 00 00 | 0014 00 00 00 00 00 | 03 00
0036 00 00 00 00 00 00 | 0018 00 00 00 00 00 | 02 00
Address cannot be reached | s ms mcs | 01 00
CMD > | 000 000 004 | 00 00
4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓

```

Вилучити це попередження і повторити спробу введення команди вже з правильною адресою можливо переміщенням курсору лівіше першої букви (символу) введеної команди (і тим самим замінити помилкову команду на попередню).

При асемблюванні команд програми підтримуються імена регістрів спеціального призначення. Чисельні операнди мають бути подані у шістнадцятковому форматі без зазначення букви “H” і не повинні починатися з букви (за необхіднос-

ті чисельні значення доповнюють лідируючим нулем, наприклад, “0A7”, “0C”, “0FF”).

Для виходу з режиму АСЕМБЛЕРА натисніть клавішу [Q] або виведіть мерехтливий курсор з поля поточної інструкції:

0000	E5F0	MOV	A,B
0002	00	NOP	
0003	00	NOP	
0007	00	MOV	

4.3. Робота з точками переривання

Меню точок переривання викликається клавішею [F5]. При її натисненні у верхній частині головного вікна налагоджувача з’являється вікно точок переривання:

Breakpoint Menu			
BRN.	PC	Counter	Occur.
1	0000	000	000
2	0000	000	000
3	0000	000	000
4	0000	000	000
5	0000	000	000
6	0000	000	000
7	0000	000	000
8	0000	000	000

Одночасно можна встановити до 8 точок переривання. Зупинка набраної і запущеної на виконання програми відбувається при досягненні зазначеної у колонці “PC” адреси, якщо кількість проходів через зазначену адресу дорівнює заданій у колонці “Counter” кількості. Тобто, зупинка виконання програми здійснюється за умови “Counter”=“Occur”. “Counter” – це лічильник, вміст якого визначає, скільки разів програма повинна пройти через названу адресу, щоб сталася зупинка. “Occur” показує, скільки разів асемблерна програма проходила через зазначену адресу.

Закінчивши редагування, натисканням клавіші [F2] необхідно зберегти на диску поточні значення точок переривання. При цьому у користувача запитується номер точки переривання (0...9):

```

000  |  you may save breakpoints in file up to 10 sets
000  |  of breakpoints (F2) and restore them (F3).
      |  Enter breakpoint set number (0-9) >
C    |  00C4 00 00 00 00 00 00 00 | C AC F0 S1 S0 0V ** P
      |  0000 00 00 00 00 00 00 00 | 0 0 0 0 0 0 1 0

```

Оскільки може бути встановлено декілька точок переривання, то слід вибрати необхідний для роботи номер точки переривання і натиснути відповідну клавішу (наприклад, 3 коли введено не менше трьох точок переривання). При цьому запит зникає, а інформація записується до файлу під іменем fd51.brk.

За необхідності змінити вибраний номер набору точок переривання слід натиснути клавішу [F1]. Це викличе повторний запит номеру точки переривання і, таким чином, його можна змінити. Для повернення до головного вікна слід натиснути [F5]. Визначивши точки переривання, запускайте програму командою G без параметрів запуску:

```

0014 00 00 00 00 00 00 00 | 0014 00 00 00 00 00 | LL 00
00FA 00 00 00 00 00 00 00 | 0018 00 00 00 00 00 | ED 00
      | Executing. Press any key to stop mcs | EC 00
CMD > | 000 292 689 | EB 00

```

```

4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓

```

При зупинці програми за перериванням видається повідомлення із зазначенням номера точки переривання:

```

0030 00 00 00 00 00 00 00 | 0014 00 00 00 00 00 | 03 00
0036 00 00 00 00 00 00 00 | 0018 00 00 00 00 00 | 02 00
      | Stopped at Breakpoint 3 | s ms mcs | 01 00
CMD > | 000 000 005 | 00 00

```

```

4 In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm ↑ 9 I/E ↓ 0 Pgm ↓

```

4.4. Функціональні клавіші та команди налагоджувача

Керування налагоджувачем здійснюється за допомогою функціональних клавіш та керуючих команд. У нижній частині головного вікна налагоджувача знахо-

диться рядок меню, який свідчить про призначення функціональних клавіш:

```
000A 00    NOP          | CMD >_          | 000 000 000 | 00 00 |
000B 00    NOP
1 Step  2 Proc  3Reg:DEC  4 In/Ex  5SetBrk  6Mem:BIN  7 I/E ↑  8 Pgm ↑  9 I/E ↓  0 Pgm ↓
```

- **1Step** – [F1] – виконати поточну інструкцію набраної асемблерної програми. Поточна інструкція виділена у вікні редагування тексту програми світлим блакитним прямокутником (див. п. 4.2). Після натиснення клавіші [F1] на екрані одразу можна побачити результати виконання поточної інструкції;
- **2Proc** – [F2] – виконати програму до наступної за вибраною адресою інструкції. Ця клавіша дає можливість *виконати підпрограму або цикл як одну інструкцію*. Отже, вже налагоджені підпрограми можна не переглядати;
- **3Reg:DEC** – [F3] — подати чисельну інформацію на екрані (вміст регістрів і пам'яті) у десятковій, а при повторному натисненні – у двійковій та шістнадцятковій формах (BIN, HEX). Після запуску налагоджувача числові дані – у шістнадцятковому вигляді;
- **4In/Ex** – [F4] – перемикає вікно пам'яті даних з внутрішньої (Internal RAM) на зовнішню (External RAM) і навпаки;
- **5SetBrk** – [F5] — встановлення точок переривання;
- **6Mem:BIN** – [F6] – подати інформацію у вікні пам'яті даних у двійковій (BIN), а при повторному натисненні - у шістнадцятковій формі (HEX);
- **7I/E ↑** – [F7] – перегорнути вікно пам'яті даних угору на один рядок;
- **8PGM ↑** – [F8] – перегорнути вікно постійної пам'яті програм угору на один рядок;
- **9I/E ↓** – [F9] – перегорнути вікно пам'яті даних униз на один рядок;
- **0Pgm ↓** – [F10] – перегорнути вікно постійної пам'яті програм униз на один рядок.

Для швидкого перегортання пам'яті можна користуватися клавішами:

- [Home] – перегорнути вікно пам'яті даних угору на одну сторінку;
- [End] – перегорнути вікно пам'яті даних униз на одну сторінку;
- [Pg Up] – перегорнути вікно пам'яті програм угору на одну сторінку;
- [Pg Dn] – перегорнути вікно пам'яті програм униз на одну сторінку.

Як вже було зазначено, керувати процесом налагодження асемблерної програми можна також і за допомогою команд, які набираються у полі командного рядка:



У цих методичних вказівках будемо застосовувати такі позначки:

- параметри команд налагоджувача-симулятора вміщені до кутових дужок, наприклад: <адреса>;
- необов'язкові параметри команд вміщені до квадратних дужок, наприклад: [<адреса>].

При введенні числових даних слід дотримуватися шістнадцяткового формату. Буква “H” для зазначення формату при цьому не ставиться. Команди можна набирати як великими, так і малими літерами.

Увага! Під час набору команд слід використовувати символи пропуску [space], які ігноруються налагоджувачем і не впливають на результати його роботи.

Програмний налагоджувач-симулятор FD51 підтримує наступні найбільш уживані команди:

R<номер регістра>=<число>

Занести число до зазначеного регістра поточного банку регістрів. Число повинне бути байтом. Приклади: R4=A7; R5=3; R1=0. Для виконання набраної у командному рядку команди натискають клавішу **[Enter]**.

Невірно набрана інструкція налагоджувачем ігнорується і він готовий до повторного введення нової команди).

<Ім'я регістра>=<число>

Занести число до регістра спеціального призначення. Можна використовувати імена всіх спеціальних регістрів (за винятком PSW), а саме: A, B, TH0, TH1, TL0, TL1, DPH, DPL, DPTR, SP, IP, IE, TMOD, TCON, SCON, SBUF, PC.

Для регістрів DPTR і PC число може бути двобайтним. Приклади: SP=31; PC=A3D; DPTR=FF47.

PO<номер порту>=<число>

Занести число до порту. Номер порту може бути у межах 0...3. Число повинне бути байтом. Приклади: PO3=25; PO1=D7; PO0=0.

<Тип пам'яті> <адреса>[-<кінцева адреса>]=<число>

Занести число до елемента пам'яті за зазначеною адресою. Якщо задана кінцева адреса, цим числом заповнюється область пам'яті. Для зазначення типу пам'яті використовуються символи "І" (*internal*, внутрішній) – внутрішня пам'ять даних; "Е" (*external*, зовнішній) – зовнішня пам'ять даних; "Р" (*program*) – пам'ять програм. Приклади: Е 27=58; І 36-40=А7; Р 31-59=0.

Увага! При заповненні деяких елементів пам'яті можливе виникнення неоднозначностей, викликаних тим, що написання мнемоніки команди і адреси елемента пам'яті може збігатися з назвою регістра. Наприклад, команда "Р С=23" буде прийнята налагоджувачем не як команда занесення числа 23 до елемента пам'яті програм за адресою 0С, а як команда запису числа 23 до програмного лічильника РС. Для попередження помилок числа, що починаються з букви, бажано передувати нулем: Р 0С=23.

<Ім'я регістра>. <номер біта> = <число>

Встановити або скинути зазначений біт регістра спеціального призначення А, В, Р00–Р03, ІР, ІЕ, ТМ0D, ТС0N, SС0N. Номер біта повинен належати діапазону 0...7. Якщо число = 0, відповідний біт скидається, інакше – встановлюється в одиничний стан. Приклади: А.7=0; В.5=0; Р0 3.2=1; Р0 0.7.

<Тип пам'яті> <адреса>. <номер біта> = <число>

Встановити або скинути зазначений біт елемента пам'яті. Номер біта повинен належати діапазону 0...7. Приклади: І 32.5=1; Е 21.0=0; Р 0.3=1.

<Ім'я прапорця> = <число>

Встановити чи скинути прапорець слова стану програми PSW. Імена прапорців: С, АС, F0, S1, S0, OV, P. Якщо число дорівнює 0, то прапорець скидається, інакше – встановлюється в одиничний стан. Приклади: С=1; OV=0; P=1.

INT<0/1> = <число>

Імітування високого або низького рівня на входах Int0 або Int1 запитів на переривання. Число має бути двійковим – 0 або 1. Приклади: INT0=0; INT0=1; INT1=1.

D<адреса>

Встановити адресу дизасембльованого тексту у вікні редагування тексту програми. Приклади: D250; DA76; D0.

M<тип пам'яті> <початкова адреса>

Встановити початкову адресу пам'яті у вікні, що висвітлює вміст пам'яті зазначеного типу. Приклади: M I 20; M E F3E; M E 351; M P 0.

S<тип пам'яті> <початкова адреса> <кінцева адреса>, <спец. файлу>

Зберегти вміст області пам'яті зазначеного типу у дисковому файлі. Специфікація файлу, як правило, повинна вміщувати шлях до файлу та його ім'я (у загальному випадку допускається будь-яка коректна у MS-DOS специфікація, наприклад, COM1 або LPT). Параметри <початкова адреса> і <кінцева адреса> визначають початок і кінець області пам'яті, що записується на диск. Приклади: S P 10-532, A:\FILE1; S E 0-15, E:\INTEL\INSIDE.DAT; S P 12-4A6, LPT.

Якщо у специфікації файлу зазначені або помилкові імена директорій та логічних дисків відсутні, видається повідомлення про неможливість відкриття відповідного диска чи директорія і створення файлу:

```

0000 00 00 00 00 00 00 00      0010
Cannot open
CMD >_

```

```

In/Ex 5SetBrk 6Mem:BIN7 I/E ↑ 8 Pgm

```

L[<тип пам'яті> <початкова адреса>,<спец. файлу>[/A]

Завантажити файл до пам'яті. Тип пам'яті визначається символами "I", "E" або "P", відповідно до цього файл завантажується до внутрішньої (Int), зовнішньої (Ext) або програмної (Pgm) пам'яті. Початкову адресу і тип пам'яті зазначити тільки при завантаженні чистого двійкового коду. При завантаженні файлу, виробленого ISIS-II MACRO-ASSEMBLER-ом, треба позначити тільки специфікацію файлу і ключ /A. Приклади: L P 256,A:\FILE1; L A:\PROG/A; L E 31,C:\ATMEL\DATE3.

Якщо при введенні команди зроблено помилку, то симулятор реагує видачею відповідного застереження, наприклад, «Invalid address» тощо. Для виправлення ситуації слід команду ввести повторно (вже без помилки).

PRT<тип пам'яті> <почат. адр.>—<кінц. адр.>[,<спец. файлу>]

Команда призначена для організації роздрукування у шістнадцятковому форматі дампу пам'яті, обмеженого початковою і кінцевою адресами (при цьому специфікація файлу повинна закінчуватися розширенням .dat).

Приклади: PRT E 27-59,Z:\abc.dat; PRT I 0-43,E:\vvc.dat.

Для роздрукування збереженого файлу (наприклад, abc.dat) слід за допомогою провідника відшукати його на диску Z та виділити курсором. Відкривати його слід за допомогою програми «Блокнот». Відкритий таким чином файл можна переглянути або роздрукувати.

PRTD<почат. адр.>,<кількість команд>[,< спец. файлу>]

Команда призначена для організації роздрукування тексту програми у мнемонічних кодах, розпочинаючи з адреси, зазначеної параметром <почат. адр.> (при цьому специфікація файлу повинна закінчуватися розширенням .txt).

Приклади: PRTD 33,20,A:\DRIVE.txt; PRTD 12,97,Z:\mmm.txt.

Для роздрукування збереженого файлу (наприклад, mmm.txt) слід за допомогою провідника відшукати його на диску Z та виділити курсором. Відкривати його слід за допомогою програми «Блокнот». Відкритий таким чином файл можна переглянути, скорегувати або роздрукувати.

G[<початкова адреса>[,<кінцева адреса>]]

Виконати програму з початкової адреси до кінцевої. Якщо початкову адресу не зазначено, програму розпочинати з поточної команди, виділеної світлим блакитним прямокутником. Кінцеву адресу можна опустити, якщо використовуються точки переривання. Можна також подати тільки кінцеву адресу, але кома у форматі при цьому повинна бути присутня. Виконання програми можна зупинити натисканням будь-якої клавіші. Запустити програму на виконання без параметрів можна за допомогою сполучення клавіш [Alt]+[F10].

Приклади: G50-91; G; G36; G,13A7.

T ON [,<спец. файлу>]

Ввімкнути трасування програми для отримання результатів по-командного виконання програми (наприклад, запущеної на виконання командою **G**) із показом вмісту всіх регістрів поточного банку даних та регістрів спеціального призначення (якщо специфікація файлу не зазначена, повідомлення трасування виводяться на принтер).

Приклади: T ON; T ON, A:\ TRACE.DAT.

Результати виконання програми (наприклад, запущеної на виконання командою **G**) можна переглянути у файлі, наприклад, TRACE.DAT, проаналізувати та роздрукувати їх за допомогою програми «Блокнот» (як і при користуванні командою **PRT**).

T OFF

Вимкнути трасування програми.

RSTC

Скинути лічильник часу виконання програми:

RST

Імітація скидання мікроконтролера.

N

Скидання всіх даних налагоджувача FD51.

H

Вивести довідкову інформацію з команд та функціональних клавіш налагоджувача. Для виходу із вікна налагоджувача натисніть клавішу [Esc].

QUIT

Вихід до DOS/Windows.

Запитання для самоперевірки

- Які дії необхідно виконати для переведення налагоджувача у режим повноекранного редагування?
- Як позначається поточна інструкція асемблерної програми у вікні налагоджувача?
- Чим відрізняються команди налагоджувача I27=55 і I2_7=55 (де "_" – символ пропуску)?
- Чим відрізняються команди P_C=31 і P_0C=31 ? Які дії виконують ці команди?
- Чи обов'язково набирати команди налагоджувача великими літерами ?
- Яку дію виконують команди P0.7=1; P0.7=3 і A.3=5 ? У чому їх особливість?
- Про що свідчить надпис "R1=05→31" у верхньому лівому куті головного вікна налагоджувача?
- Чим відрізняються дії функціональних клавіш [F1] і [F2] налагоджувача?
- Чим відрізняються дії команд RST і N налагоджувача?
- Яку інформацію можна отримати, переглянувши файл трасування програми?

Контрольне завдання для самоперевірки рівня опанування навичок роботи з симулятором

Для самоперевірки рівня опанування навичок роботи з симулятором-налагоджувачем слід виконати самостійно наступну послідовність дій:

- запустити налагоджувач-симулятор із розгорнутим на увесь екран головним вікном;
- опанувати операції перегляду та змінювання вмісту елементів пам'яті даних і програм, реєстрів спеціальних функцій, портів та байта стану, а також окремих бітів у ручному режимі та режимі командного рядка;
- опанувати операції введення контрольних точок програми, розташованої у пам'яті розпочинаючи з адреси 0000H, наступного вигляду:

```
MOV    A,#00 ; скид вмісту акумулятора в нуль
INC    A      ; збільшення вмісту акумулятора на 1
AJMP   0002   ; перехід на виконання інструкції INC A
```

так, що при зупинці виконання програми в акумуляторі А у підсумку було число, яке відповідає порядковому номеру студента у навчальному журналі;

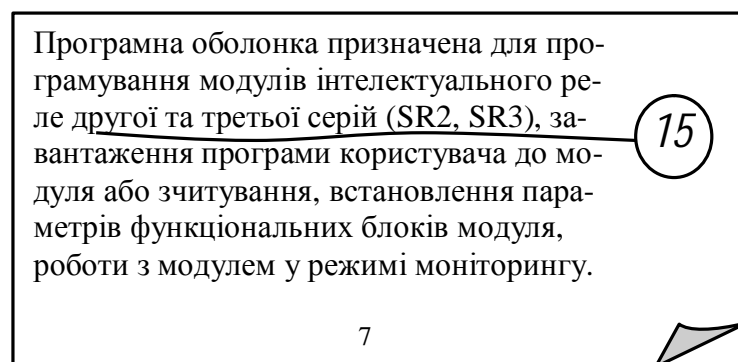
- опанувати операції запису та читання програми на вибраний диск персональної ЕОМ;
- перевірити виконання та результати роботи решти не зазначених команд мікроконтролера.

Індивідуальні завдання до самостійної роботи

Завдання полягає у опрацюванні методичних вказівок з олівцем у руках.

При виконанні завдання необхідно відшукувати та підкреслювати фрагменти тексту або позначати на ілюстраціях графічні елементи, про які йдеться у запитаннях.

Поруч, на березі аркуша методичних вказівок, у кружечку слід проставити номер запитання, наприклад:



Також у графі «Посилання» треба зазначити номер сторінки, на якій знайдено відповідь на запитання.

Увага! Виконання наведених завдань дозволяється лише у власному примірнику методичних вказівок. Забороняється проставляти будь-які позначки або виконувати креслення у методичних вказівках бібліотечного фонду.

Індивідуальні варіанти завдань до самостійної роботи

№ зап.	№ вар.	Завдання або запитання	Поси- дання
1	1/1	Підкресліть фрагмент тексту методичних вказівок, у якому йдеться про найбільш суттєві відмінності архітектури однокристальних мікроконтролерів від традиційної архітектури мікропроцесорів.	
2	2/2	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр-розширювач акумулятора.	
3	3/3	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр режимів таймерів-лічильників.	
4	4/4	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр управління/статусу таймерів	
5	5/5	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр пріоритетів переривань.	
6	6/6	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр маски переривань.	
7	7/7	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр управління електроспоживанням.	
8	8/8	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр управління приймачем-передавачем.	
9	9/9	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, буфер приймача-передавача.	
10	10/10	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, сигнал стробування адреси зовнішньої пам'яті.	
11	11/11	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, сигнал дозволу зовнішньої пам'яті програм.	
12	12/12	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, сигнал дозволу внутрішньої пам'яті програм.	
13	13/13	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, сигнал загального скидання мікроконтролера.	
14	14/14	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, виводи для підключення кварцового резонатора.	
15	15/15	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр, в якому зберігається прапорець ознаки перенесення.	
16	1/16	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр, в якому зберігається прапорець ознаки допоміжного перенесення.	
17	2/17	Позначте на рисунку, що відображує внутрішню архітектуру мікроконтролера, реєстр, в якому зберігається прапорець ознаки паритету.	

18	3/18	Позначте на рисунку, що відображає внутрішню архітектуру мікроконтролера, регістр, в якому зберігаються прапорці селектора банків регістрів загального призначення.	
19	4/19	Позначте на рисунку, що відображає розташування виводів мікроконтролера, лінії, якими передається молодший байт адреси при звертанні до зовнішньої пам'яті.	
20	5/20	Позначте на рисунку, що відображає розташування виводів мікроконтролера, лінії, якими передається старший байт адреси при звертанні до зовнішньої пам'яті.	
21	6/1	Позначте на рисунку, що відображає розташування виводів мікроконтролера, лінії, якими передається або приймається байт даних при звертанні до зовнішньої пам'яті.	
22	7/2	Позначте на рисунку, що відображає розташування виводів мікроконтролера, сигнал стробування адреси зовнішньої пам'яті.	
23	8/3	Позначте на рисунку, що відображає розташування виводів мікроконтролера, сигнал дозволу внутрішньої пам'яті програм.	
24	9/4	Позначте на рисунку, що відображає розташування виводів мікроконтролера, сигнал дозволу зовнішньої пам'яті програм.	
25	10/5	Позначте на рисунку, що відображає розташування виводів мікроконтролера, сигнал загального скидання мікроконтролера.	
26	11/6	Позначте на рисунку, що відображає розташування виводів мікроконтролера, виводи для підключення кварцового резонатора.	
27	12/7	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вхід послідовного порту.	
28	13/8	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вихід послідовного порту.	
29	14/9	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вхід зовнішнього переривання 0.	
30	15/10	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вхід зовнішнього переривання 1.	
31	1/11	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вхід таймера-лічильника 0.	
32	2/12	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вхід таймера-лічильника 1.	
33	3/13	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вихід строб-сигналу запису до зовнішньої пам'яті даних.	
34	4/14	Позначте на рисунку, що відображає розташування виводів мікроконтролера, вихід строб-сигналу зчитування із зовнішньої пам'яті даних.	
35	5/15	Позначте на рисунку, що відображає розташування виводів мікроконтролера, виводи для підключення напруги живлення мікроконтролера.	
36	6/16	Позначте у головному вікні налагоджувача FD51 регістри загального призначення банку 0, які можна використовувати для непрямой (посередньої) адресації.	
37	7/17	Позначте у головному вікні налагоджувача FD51 регістри загального призначення банку 1, які можна використовувати для непрямой (посередньої) адресації.	
38	8/18	Позначте у головному вікні налагоджувача FD51 регістри загального призначення банку 2, які можна використовувати для непрямой (посередньої) адресації.	

39	9/19	Позначте у головному вікні налагоджувача FD51 реєстри загального призначення банку 3, які можна використовувати для непрямой (посередньої) адресації.	
40	10/20	Позначте у головному вікні налагоджувача FD51 елементи пам'яті, що адресуються реєстрами R0 та R1 нульового банку реєстрів.	
41	11/1	Позначте у головному вікні налагоджувача FD51 елементи пам'яті, що адресуються реєстрами R0 та R1 першого банку реєстрів.	
42	12/2	Позначте у головному вікні налагоджувача FD51 елементи пам'яті, що адресуються реєстрами R0 та R1 другого банку реєстрів.	
43	13/3	Позначте у головному вікні налагоджувача FD51 елементи пам'яті, що адресуються реєстрами R0 та R1 третього банку реєстрів.	
44	14/4	Позначте у головному вікні налагоджувача FD51 у вікні INT RAM реєстр R3 нульового банку реєстрів.	
45	15/5	Позначте у головному вікні налагоджувача FD51 у вікні INT RAM реєстр R0 першого банку реєстрів.	
46	1/6	Позначте у головному вікні налагоджувача FD51 у вікні INT RAM реєстр R0 другого банку реєстрів.	
47	2/7	Позначте у головному вікні налагоджувача FD51 у вікні INT RAM реєстр R0 третього банку реєстрів.	
48	3/8	Позначте у головному вікні налагоджувача FD51 у колонці Bank 0 елемент внутрішньої пам'яті даних з адресою 06h.	
49	4/9	Позначте у головному вікні налагоджувача FD51 у колонці Bank 1 елемент внутрішньої пам'яті даних з адресою 0Ch.	
50	5/10	Позначте у головному вікні налагоджувача FD51 у колонці Bank 2 елемент внутрішньої пам'яті даних з адресою 16h.	
51	6/11	Позначте у головному вікні налагоджувача FD51 у колонці Bank 3 елемент внутрішньої пам'яті даних з адресою 1Dh.	
52	7/12	Позначте у головному вікні налагоджувача FD51 у вікні редагування тексту програми колонку з поточними адресами інструкцій (команд) програми.	
53	8/13	Позначте у головному вікні налагоджувача FD51 у вікні редагування тексту програми колонку з машинними кодами інструкцій (команд) програми.	
54	9/14	Позначте у головному вікні налагоджувача FD51 у вікні редагування тексту програми колонку з мнемонічними позначеннями інструкцій (команд) програми.	
55	10/15	Позначте у головному вікні налагоджувача FD51 прапорець ознаки перенесення.	
56	11/16	Позначте у головному вікні налагоджувача FD51 прапорець ознаки допоміжного перенесення.	
57	12/17	Позначте у головному вікні налагоджувача FD51 прапорець ознаки паритету.	
58	13/18	Позначте у головному вікні налагоджувача FD51 прапорець ознаки переповнення.	
59	14/19	Позначте у головному вікні налагоджувача FD51 прапорець користувача слова стану програми.	
60	15/20	Позначте у головному вікні налагоджувача FD51 прапорці селектора банків реєстрів.	
61	1/1	Позначте у головному вікні налагоджувача FD51 старший байт реєстра DPTR.	

62	2/2	Позначте у головному вікні налагоджувача FD51 молодший байт реєстра DPTR.	
63	3/3	Позначте у головному вікні налагоджувача FD51 старший байт реєстра програмного лічильника.	
64	4/4	Позначте у головному вікні налагоджувача FD51 молодший байт реєстра програмного лічильника.	
65	5/5	Позначте у головному вікні налагоджувача FD51 старший напівбайт (старшу тетраду) реєстра режимів таймерів-лічильників.	
66	6/6	Позначте у головному вікні налагоджувача FD51 молодший напівбайт (молодшу тетраду) реєстра режимів таймерів-лічильників.	
67	7/7	Позначте у головному вікні налагоджувача FD51 старший напівбайт (старшу тетраду) реєстра управління/статусу таймерів-лічильників.	
68	8/8	Позначте у головному вікні налагоджувача FD51 молодший напівбайт (молодшу тетраду) реєстра управління/статусу таймерів-лічильників.	
69	9/9	Позначте у головному вікні налагоджувача FD51 старший напівбайт (старшу тетраду) реєстра пріоритетів переривань.	
70	10/10	Позначте у головному вікні налагоджувача FD51 молодший напівбайт (молодшу тетраду) реєстра пріоритетів переривань.	
71	11/11	Позначте у головному вікні налагоджувача FD51 старший напівбайт (старшу тетраду) реєстра маски переривань.	
72	12/12	Позначте у головному вікні налагоджувача FD51 молодший напівбайт (молодшу тетраду) реєстра маски переривань.	
73	13/13	Позначте у головному вікні налагоджувача FD51 старший напівбайт (старшу тетраду) реєстра керування приймачем-передавачем.	
74	14/14	Позначте у головному вікні налагоджувача FD51 молодший напівбайт (молодшу тетраду) реєстра керування приймачем-передавачем.	
75	15/15	Позначте у головному вікні налагоджувача FD51 елемент пам'яті програм з адресою 0Dh.	
76	1/16	Позначте у головному вікні налагоджувача FD51 елемент пам'яті програм з адресою 11h.	
77	2/17	Позначте у головному вікні налагоджувача FD51 елемент пам'яті програм з адресою 15h.	
78	3/18	Позначте у головному вікні налагоджувача FD51 елемент пам'яті програм з адресою 1Ah.	
79	4/19	Позначте у головному вікні налагоджувача FD51 елемент пам'яті програм з адресою 0Ah.	
80	5/20	Позначте у головному вікні налагоджувача FD51 командний рядок для введення команд налагоджувача.	
81	6/1	Позначте у головному вікні налагоджувача FD51 команду меню для покрокового виконання програми.	
82	7/2	Позначте у головному вікні налагоджувача FD51 команду меню для перемикання вікна пам'яті даних з внутрішньої на зовнішню та навпаки.	
83	8/3	Позначте у головному вікні налагоджувача FD51 команду меню для встановлення точок переривання.	
84	9/4	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду занесення числа до зазначеного реєстра поточного банку реєстрів.	
85	10/5	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду занесення числа до зазначеного реєстра спеціального призначення.	
86	11/6	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду занесення числа до зазначеного порту.	

87	12/7	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду занесення числа до елемента пам'яті за зазначеною адресою.	
88	13/8	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, якою можна встановити біт регістра спеціального призначення.	
89	14/9	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, якою можна скинути біт регістра спеціального призначення.	
90	15/10	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, якою можна встановити біт елемента пам'яті.	
91	1/11	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, якою можна скинути біт елемента пам'яті.	
92	2/12	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду імітування високого або низького рівня напруги на входах запитів на переривання.	
93	3/13	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, що встановлює бажану адресу у вікні редагування тексту програми (вікні дизасемблера).	
94	4/14	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, що встановлює бажану початкову адресу у вікні пам'яті зазначеного типу.	
95	5/15	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду для збереження вмісту області пам'яті зазначеного типу у файлі.	
96	6/16	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду завантаження файлу до пам'яті зазначеного типу.	
97	7/17	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду для роздруку (дампінгу) області пам'яті зазначеного типу.	
98	8/18	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду для роздруку програми у мнемонічних кодах.	
99	9/19	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду запуску програми у режимі прогону.	
100	10/20	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду увімкнення режиму трасування програми.	
101	11/1	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду вимкнення режиму трасування програми.	
102	12/2	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, що дозволяє скинути лічильник часу виконання програми.	
103	13/3	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, що дозволяє імітувати апаратне скидання мікроконтролера.	
104	14/4	Знайдіть та підкресліть у переліку команд налагоджувача FD51 команду, що дозволяє скинути усі дані налагоджувача.	

Увага ! Після виконання завдань самостійної та контрольної роботи студент допускається до тестових випробувань за допомогою спеціальної тестової програми на персональній ЕОМ. У випадку не виконання тестових завдань робота вважається не захищеною.

РЕСУРСИ ВСЕСВІТНЬОЇ МЕРЕЖІ ІНТЕРНЕТ

- **Інформаційний сервер “ЧипИнфо”.** Довідкова інформація щодо будь-яких електронних компонентів.
 - <http://www.chipinfo.ru>
- **Інформаційний сервер “Российский рынок микроэлектроники».** Довідкова інформація щодо будь-яких електронних компонентів.
 - <http://www.gaw.ru>
- **Motorola.** Мікроконтролери та електронні компоненти:
 - <http://Design-NET.com>
- **Silicon Labs.** Провідний виробник мікроконтролерів MCS-51.
 - <http://www.silabs.com>
- **Російський сервер сектора напівпровідникових компонентів компанії Motorola:**
 - <http://www.motco.ru>
- **Zilog, Inc.** Мікропроцесори, мікроконтролери та електронні компоненти:
 - <http://www.zilog.com>
- **Scenix Semiconduktor, Inc.** Мікроконтролери та електронні компоненти:
 - <http://www.scenix.com>
- **Parallax, Inc.** Засоби розробки для мікроконтролерів SX фірми Scenics:
 - <http://www.parallaxinc.com>
- **Holtek.** Мікроконтролери та електронні компоненти:
 - <http://www.holtek.com.tw>
- **Atmel Corp.** MCS-51-сумісні мікроконтролери, RISC-мікроконтролери, та електронні компоненти:
 - <http://www.atmel.com>, <http://www.atmel.ru>
- **Intel.** Мікропроцесори, мікроконтролери та електронні компоненти:
 - <http://www.intel.com>
- **КТЦ-МК.** Консультаційно-технічний центр з мікроконтролерів:
 - <http://www.cec.ru>
- **ФИТОН.** Програмні та апаратні засоби розробки для мікроконтролерів PIC, MCS-51, MCS-96 та їх аналогів
 - <http://www.phyton.ru>
- **ВАТ «НИИМЭ и Микрон».** Розробка та виробництво мікросхем логіки, ЦАП, АЦП, ОЗП, ПЗП, контролерів побутових пристроїв та ін.:
 - <http://www.mikron.ru>
- **Інтернет-книгарня «Книги России»:**
 - <http://www.books.ru>

- **КВАЗАР-Мікро.** Мікропроцесори, мікроконтролери, інтегральні схеми та напівпровідникові компоненти. Офіційний дистриб'ютор Intel Corporation на Україні:
- <http://www.kvazar-micro.com>

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Бродин В.Б., Шагурин И.И. Микроконтроллеры. Архитектура, программирование, интерфейс. – М.: ЭКОМ, 1999. – 400 с.
2. Встраиваемый микроконтроллер 8XC251SB: Руководство пользователя: Пер. с англ. – К.: Квазар-микро, 1995. – 418 с.
3. Гуртовцев А.Л., Гудыменко С.В. Программы для микропроцессоров: Справ. пособие. – Минск: Вышейш. шк., 1989. – 352 с.
4. Долгий А. Разработка и отладка устройств на МК // Радио. – 2001. – №№ 5-10. – С. 14-16, с. 17-19.
5. Злобин В.К., Григорьев В.Л. Программирование арифметических операций в микропроцессорах. – М.: Высш. шк., 1991. – 303 с.
6. Каспер Э. Программирование на языке Ассемблере для микроконтроллеров семейства i8051. – М.: Горячая линия – Телеком, 2003. – 191 с.
7. Николайчук О.И. x51 – совместимые микроконтроллеры фирмы Signal. – М.: ИД СКИМЕН, 2002. – 230 с.
8. Однокристалльные микроЭВМ: Справочник / А.В. Бобрыкин, Г.П. Липовецкий, Г.В. Литвинский и др. – М.: МИКАП, 1994. – 400 с.
9. Современные микроконтроллеры: Архитектура, средства проектирования, примеры применения, ресурсы сети Интернет / Под ред. И.В. Коршуна; Составление, пер. с англ. Б.Б. Горбунова. – М.: Аким, 1998. – 272 с.
10. Фрунзе А.В., Фрунзе А.А. Микроконтроллеры? Это же просто! Т. 3. – М.: ИД СКИМЕН, 2003. – 224 с
11. Lopez S. Using the 87C51GB. AB-44 Application Brief. March 1991. Order Number: 270957-001. Intel Corporation, 1996, p. 1-18, A.1-A.7, B.1.
12. Matthew Chapman. THE FINAL WORD ON THE 8051, 1994, 255 p.
13. MCS®-51 and MCS®-96 Packaging Information. November 1994. Order Number: 272118-001. Intel Corporation, 1995, 16 p.
14. Schue R. 32-Bit Math Routines for the 8051. AB-40 Application Brief. October 1992. Order Number: 270530-002. Intel Corporation, 1996, 8 p.
15. 8XC251SA/SB/SP/SQ High-Performance CHMOS Microcontroller. Product Preview. December 1995. Order Number: 272783-002. Intel Corporation, 1995, 35 p.
16. 8XC51GB CHMOS Single-Chip 8-Bit Microcontroller. Preliminary. November 1994. Order Number: 272337-002. Intel Corporation, 1995, 22 p.

Упорядники:

КИРИЧЕНКО Віталій Іванович
ЯЛАНСЬКИЙ Олексій Анатолійович

МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ МПП-1
“АРХІТЕКТУРА МІКРОКОНТРОЛЕРА 8051АН СІМЕЙСТВА MCS-51. ПОВНОЕК-
РАННИЙ НАЛАГОДЖУВАЧ-СИМУЛЯТОР FD51”, ІНДИВІДУАЛЬНИХ ЗА-
ВДАНЬ ТА САМОСТІЙНОЇ РОБОТИ З ПРОФЕСІЙНО-ОРІЄНТОВАНОЇ ДИС-
ЦИПЛІНИ “МІКРОПРОЦЕСОРНІ ПРИСТРОЇ”

для студентів спеціальності 7.092203 “Електромеханічні системи автоматизації та еле-
ктропривід” напрямку “Електромеханіка”

Редакційно-видавничий комплекс
Друкується у редакційній обробці упорядників

Підписано до друку . Формат 30×42/4.
Папір офсет. Ризографія. Умовн. друк. арк. .
Обліково-видавн. арк. . Тираж 100 прим. Зам. №

НГУ
49027, м. Дніпропетровськ-27, просп. К. Маркса, 19.