

Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ДОМАШНЯ КОНТРОЛЬНА РОБОТА №1**

**З ДИСЦИПЛІНИ “АРХІТЕКТУРА КОМП’ЮТЕРА”**

**НАЗВА: “РОЗРОБКА ПРОГРАМ ПЕРЕДАЧІ ДАНИХ І ПРОГРАМНЕ ФОРМУВАННЯ ЧАСОВОЇ ЗАТРИМКИ В МК51”**

**Виконала:**

студентка групи ІО – 64

Кішка М. І.

**Перевірила:**

Доцент Ткаченко В. В.

м. Київ – 2018 р.

***Мета:*** Вивчення системи команд, форматів подання даних та способів адресації операндів; вивчення команд передачі управління, команд пересилки даних та команд вибору банків регістрів; отримання навиків розробки програм на мові асемблеру МК51.

**Короткі теоретичні відомості**

Загальна структурна схема мікроконтролера зображена на рис. 1. Вона має такий вигляд:

Рис. 1 Структурна схема мікроконтролеру МК51

Мікроконтролер містить резидентну пам'ять програм (РПП), ємністю 4 кбайти, призначену для зберігання команд, констант, управляючих слів ініціалізації, таблиць кодування вхідних і вихідних змінних, та резидентну пам'ять даних (РПД), призначену для зберігання змінних у процесі виконання програми, адресується одним байтом і має ємність 128 байт; пристрій управління і синхронізації, до складу якого входить лічильник команд, регістр команд і регістр ознак; арифметико-логічний пристрій, до складу якого входить АЛБ, акумулятор і регістри, і призначений для виконання арифметичних і логічних операцій, і оброки інформації; блок таймерів-лічильників та блок послідовного інтерфейсу і переривань. Обмін даними здійснюється через чотири порти Р0, Р1, Р2, Р3, або через послідовний порт.

Вбудована пам'ять в мікроконтролері:

* **Резидентна пам’ять програм**, має ємність 4Кб. Призначена для зберігання команд, констант, управляючих слів ініціалізації, таблиць кодування вхідних і вихідних змінних. Резидентна пам’ять даних підключена до шістнадцятибітної шини адреси, що надходить з лічильника команд, або регістру покажчика даних.
* **Резидентна пам’ять даних** призначена для зберігання змінних у процесі виконання програми, адресується одним байтом і має ємність 128 байт. До адресного простору резидентної пам’яті даних належать регістри спеціальних функцій.

Зовнішня пам'ять може бути підключена двох видів:

* **Зовнішня пам'ять програм** реалізується за допомогою додаткових мікросхем пам'яті і може працювати лише за рахунок встановлення сигналу EMA у 0. Під час звернення до зовнішньої пам'яті програм старший байт адреси передається через порт Р2. Видача молодшого байта адреси, а також і передавання байта даних здійснюється через порт Р0 в режимі часового мультиплексування.
* **Зовнішня пам'ять даних** реалізується за допомогою додаткових мікросхем пам'яті і може мати ємність до 64 Кб з адресами 0 –FFFFh. Адресний простір резидентної і зовнішньої пам'яті даних не перетинаються, оскільки доступ до зовнішньої пам'яті здійснюється за допомогою спеціальних команд.

**Система команд** мікроконтролера КР1816ВЕ51 значно ширша і потужніша ніж система команд КР1816ВЕ48, за рахунок команд множення, ділення, віднімання, операцій над бітами, операцій зі стеком, розширеного набору команд передачі управління.

Система команд мікроконтролера містить *сто одинадцять команд*. Відносно функціональних ознак команди класифікуються за наступними групами:

* команди передачі даних;
* команди виконання арифметичних операцій;
* команди виконання логічних операцій;
* команди виконання операцій з бітами;
* команди передачі управління.

Команди МК51 мають довжину один, два або три байти і виконуються відповідно за один, два або чотири машинні цикли. За тактової частоти генератора fr = 12Мгц тривалість циклу складає 1 мкс.

Затримки у МК51 формуються двома шляхами: програмно, використовуючи команди управління, і апаратно, застосовуючи два шістнадцятирозрядних лічильники/таймери Т/Л0 і Т/Л1. Розглянемо обидва способи детально.

Чисто програмний спосіб задавання затримок базується на використанні основних команд управління, що забезпечують повернення в значення минулої мітки в коді. Затримка формується завдяки наявності деякого часу обробки і виконання самим мікроконтролером цих команд. Час обробки команд забезпечується за допомогою машинних циклів, які формуються завдяки зовнішнього кварцового резонатора. Всі команди управління (JMP, AJMP, LJMP тощо) виконуються за 2 машинних цикли. Один машинний цикл на пристрої управління формується за 12 періодів резонатора(часових тактів). При частоті кварцового резонатора в 12 МГц, час виконання одного машинного циклу досягає 1 мкс.

Тоді, відповідно до часу виконання команд можна забезпечити затримку 1 команди управління в 2 мкс. При побудові циклу через регістри загального забезпечення можна сформувати затримки короткої тривалості.

Другий спосіб реалізації затримок в МК51 базується на апаратному використані лічильників/таймерів, які є вбудованими в мікроконтролер. Обидва Т/Л0 і Т/Л1 можуть працювати в режимах таймеру і лічильнику. В режимі таймеру відбувається інкрементація вмісту регістру таймера/лічильника із частотою f/12, де f – частота тактового генератора (внутрішнього). В режимі лічильнику інкрементація вмісту регістру відбувається на кожному переході зовнішнього сигналу з 1 на 0, що подається на вхід лічильника зовнішніх подій T1 (T0).

Управління роботами таймеру/лічильнику і підключення їх до системи переривань відбувається за рахунок регістрів спеціальних функцій TMOD і TCON.

Формат даних регістрів зображений на рис. 2:

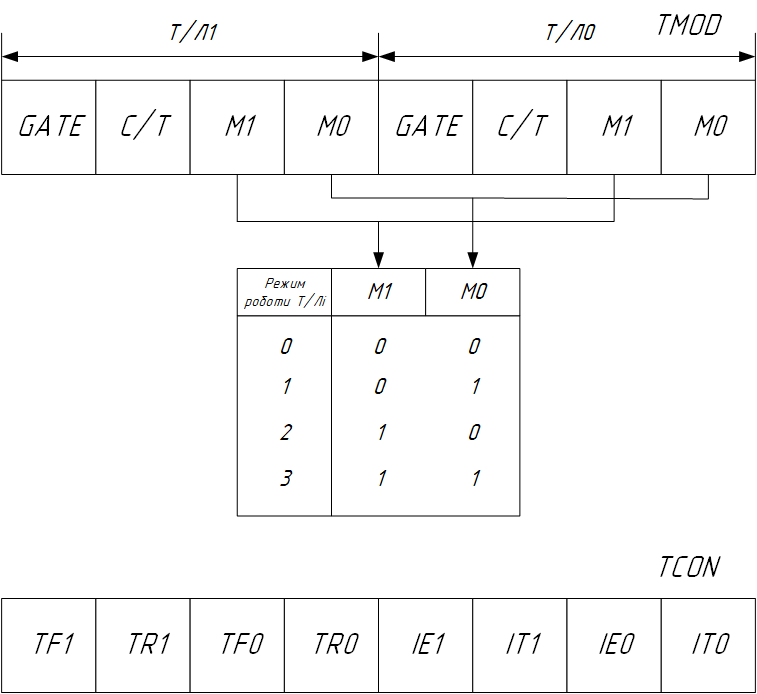


Рис. 2 Формат регістрів для роботи із Т/Лі

Позначення на форматах такі:

* GATE – управління блокуванням, за значення 1 управління можливе якщо зовнішній управляючий сигнал INTi = 1, а також біт управління TRi = 1, в іншому випадку при 0 достатньо щоб біт управління TRi був встановлений;
* C/T – біт вибору режиму, при значенні 0 визначає роботу в якості таймера від внутрішнього джерела сигналів синхронізації, при значенні 1 визначає роботу в якості лічильнику від зовнішніх сигналів на вході Ti;
* M1, M0 – визначають вибір режиму роботи таймеру/лічильника, відповідно до таблиці;
* TFi – ознака переповнення таймерів Т/Лі, встановлюються програмно, або апаратно під час переповнення Т/Лі; за включеного режиму переривання на таймерах/лічильниках встановлення ознаки створить відповідне переривання; ознаки скидаються програмно, або апаратно за обслуговування відповідного переривання;
* TRi – біти управління таймерів Т/Лі, встановлюються і скидаються програмно;
* IEi – ознаки запиту (фронту) зовнішніх переривань. Встановлюються апаратно за зрізом зовнішніх сигналів ЗПР1 (ЗПР0) або програмно; скидаються апаратно при обслуговуванні переривання, викликаного фронтом сигналу переривання;
* ITi – біти управління типом переривання, на входах INT1 і INT0; встановлюються і скидаються програмно для специфікації запиту ЗПР1 (ЗПР0); якщо ITi = 0, то дозволено переривання за низьким рівнем сиг-налу, за встановленої 1 переривання можливе за зрізом сигналу або за його низьким рівнем.

Розглянемо 4 режими роботи таймерів/лічильників, що встановлюються бітами M1 і M0:

1. За даного режиму таймер/лічильник є тринадцятирозрядним лічильником, де послідовно з’єднані п’ятирозрядний регістр TLi і восьмирозрядний регістр THi. Залежно від біту C/Ti таймер/лічильник працює або в режимі таймеру із частотою f/12, або в режимі лічильнику із зміною сигналів на вході Ti. Рахування починається за встановлення біта TRi регістра TCON. Управління рахуванням ззовні здійснюють за допомогою біту GATE регістра TMOD. При цьому рахування дозволене за встано-влення значення вхідного сигналу INTi = 1, і заборонене при його зна-чені 0. Під час переповнення Т/Лі встановлюється ознака TFi.
2. Аналогічний до режиму 0, але із різницею в тому, що Т/Лі є шістнадцятирозрядним таймером/лічильником, в якому TLi вже є восьмирозрядним.
3. За такого режиму, таймер/лічильник є восьмирозрядним на основі ре-гістру TLi. Під час кожного переповнення регістру TLi відбувається завантаження вмісту регістру THi в регістр TLi. Вміст регістру THi завантажується програмно і в процесі рахування не змінюється.
4. В цьому режимі робота Т/Л0 і Т/Л1 відрізняється. Т/Л0 працює на основі регістру TL0 і може працювати в якості і таймеру, і лічильнику. Він включений постійно, видає ознаки переповнення TF0 і може оброблювати переривання. Т/Л1 працює на основі регістру TH0 лише в ре-жимі таймеру. Він включається за встановлення біту TR1, ознака переповнення виникає на біті TF1, але цей таймер не може оброблювати переривання.

Застосовуючи таймери/лічильники можна забезпечити утворення затримок довгої тривалості, які будуть проходити апаратно на окремих регістрах.

В рамках домашньої контрольної роботи буде застосовано режим роботи 0, що відповідає режиму часових затримок в мікроконтролері МК48 із подільником частоти рівним f = 12 МГц. В такому разі, кожна наступна зміна значення регістру THi буде виконуватися за 32 мкс.

***Варіант:6410***

**Порядок виконання роботи**

**Практична робота №3.1**

1.Визначаємо загальний варіант для заданої роботи. Для цього візьмемо молодші розряди двійкового числа отриманого з номеру залікової книжки.

Отримане значення в двійковій системі:

Беремо для завдання 5 молодших розрядів цього числа: 01010.

Згідно з таблицями варіантів маємо відповідні значення для мого варіанту:

Таблиця 1. Варіант вибору ЗПП

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Кількість сторінок ПП** | **Об’єм сторінки** |
| 0 | 1 | 6 | 64 Кб |

Таблиця 2. Варіант вибору ЗПД

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Кількість сторінок ПД** | **Об’єм сторінки** |
| 0 | 1 | 6 | 64 Кб |

Таблиця 3. Варіант вибору функції для обчислення і відповідних регістрів

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Банк регістрів** | **Функція для обчислення** |
| 0 | 1 | БР1 |  |

Таблиця 4. Варіант розміру масиву, що зберігається у пам’яті

|  |  |  |
| --- | --- | --- |
|  |  | **Розмірність масиву** |
| 1 | 0 | 22 |

Таблиця 5. Варіант розміщення масиву, починаючи з якої адреси

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | **Адреса початкової комірки пам’яті** |
| 1 | 0 | 0 | 52h |

Таблиця 6. Варіант вибору запису результату

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Регістри** | **Банк регістрів** |
| 1 | 0 | R6. R7 | БР0 |

Таблиця 7. Варіант чисел для перевірки роботи програми

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ***X1*** | ***X2*** | ***X3*** | ***X4*** | ***X5*** | ***X6*** | ***X7*** | ***X8*** | ***.. X22*** |
| 0 | 1 | 12 | A2 | -11 | 0A | -23 | 11 | \* | \* | \* |

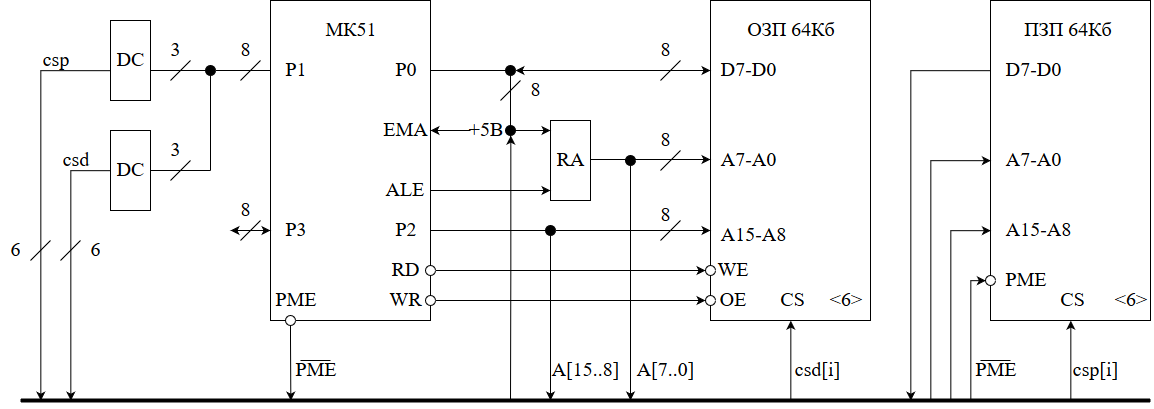
2. Виконаємо побудову операційної схеми підключення до МК51 зовнішньої пам’яті даних і програм відповідно до заданого мого варіанту. Маємо:

Рис. 3 Операційна схема підключення зовнішньої пам’яті до МК51

3. Виконаємо числове моделювання для значень отриманих із варіанту відповідно до заданого алгоритму обрахунку функції:

1)X1 = 12h, X2 = A2h; X1 v X2

Rez = 12h v A2h = B2h

2)Rez = B2h; 8\*Rez

Rez1 = RLC(RLC(RLC(B2h))) = 0590h

3)X3 = -11h, X4 = 0Ah; X3 – X4

Rez2 = -11h – 0Ah = FFEFh + FFF6h = FFE5h

4)Rez1 = 0590h, Rez2 = FFE5h; Rez1 \* Rez2

Rez3 = 0590h \* FFE5h = 69D0h

5) X5 = -23h, X6 = 11h; X5 / X6

Rez4 = -23h / 11h = FFDDh / 11h = FFFEh

6) Rez3 = 69D0h, Rez4 = FFFEh; Rez3 – Rez4

Rez5 = 69D0h - FFFEh = 69D2h

7)Rez5 = 69D2h; Rez5 / 16

Rez6 = RRC(RRC(RRC(RRC(69D2h)))) = 069Dh

R6 = 06h, R7 = 9Dh

4. На основі отриманого моделювання, виконуємо побудову програми обробки переносу масиву чисел із зовнішньої пам’яті у резидентну, а також обчислення заданого алгоритму. Використовуємо моделюючий комплекс SCM MK51.

**Лістинг програми**

;------Labwork #1--------

;JMP main

AJMP START

;------procedures--------

INPUT:

MOV PSW, #00h ;SEL RB0

MOV R0, #52h ;First cell of internal memory to save

MOV DPTR, #0052h ;First cell of external memory to read from

MOV R7, #16h ;22 elements in array

MOV P1, 00h ;1 page of OM

input1:

MOV A, @P0

MOVX @DPTR, A

INC DPTR

DJNZ R7, input1

MOV R7, #16h

MOV DPTR, #0052h

read:

MOVX A, @DPTR

MOV @R0, A

INC R0

INC DPTR

DJNZ R7, read

;moving last 6 elements into 1-6 registers

;it doesn't work from resident memory correctly

;sending it from directly p0

MOV R1, #0Eh ;address of R6, RB1

MOV R7, #06h ;6 elements

;DEC R0

read2:

;MOV A, @R0

MOV A, @P0

MOV @R1, A

;DEC R0

DEC R1

DJNZ R7, read2

INC R1

RET

FUNC:

MOV PSW, #08h ;SEL RB1

;------block 1-----------

;X1 and X2

MOV A, R1

ANL A, R2

MOV R1, A

CLR A

MOV R2, A

;------block 2-----------

;8\*X12

CPL A

MOV R7, 03h

SLC1:

MOV A, R2

CLR C

RLC A

MOV R2, A

CLR A

ORL A, R1

RLC A

MOV R1, A

DJNZ R7, SLC1

;------block 3-----------

;X3 - X4

CLR A

MOV R7, A

MOV R0, A

MOV A, R3

ANL A, #80h; 8 bit

JZ NeX4

CLR A

CPL A

MOV R0, A

NeX4:

MOV A, R4

ANL A, #80h; 8 bit

JZ next1

CLR A

CPL A

MOV R7, A

next1:

;smaller bits

CLR C

MOV A, R3

SUBB A, R4

MOV R4, A

;higher bits

MOV A, R0

SUBB A, R7

MOV R3, A

;------block 4-----------

;X12 \* X34

MOV A, R2

MOV B, R4

MUL AB

MOV R0, B

MOV R1, A

;------block 5-----------

;X5 / X6

MOV A, R5

MOV B, R6

DIV AB

MOV R5, A

MOV R6, B

;------block 6-----------

;X1234 - X56

;smaller bits

CLR C

MOV A, R1

SUBB A, R6

MOV R4, A

;higher bits

MOV A, R0

SUBB A, R5

MOV R3, A

;------block 7-----------

;X123456/16

CLR A

MOV R7, 04h

SRC1:

MOV A, R4

CLR C

RRC A

MOV R4, A

MOV A, R3

RRC A

MOV R3, A

DJNZ R7, SRC1

MOV A, R3

MOV PSW, #00h ;SEL RB0

MOV R6, A

MOV PSW, 08h ;SEL RB1

MOV A, R4

MOV PSW, #00h ;SEL RB0

MOV R7, A

RET

;------main\_program------

START:

ACALL INPUT

ACALL FUNC

END

**Практична робота №3.2**

1.Визначаємо загальний варіант для заданої роботи. Для цього візьмемо молодші розряди двійкового числа отриманого з номеру залікової книжки.

Отримане значення в двійковій системі: 0001100100001010.

Беремо для завдання 5 молодших розрядів цього числа: 01010.

Згідно з таблицями варіантів маємо відповідні значення для мого варіанту:

Таблиця 8. Варіант алгоритму для підрахування

|  |  |  |
| --- | --- | --- |
|  |  | **Алгоритм** |
| 1 | 0 |  |

Таблиця 9. Варіанти управляючих сигналів, що розташовані у операторних вершинах

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Управляючі сигнали** | | | |
| **Номер операторної вершини** | | | |
| 1 | 2 | 3 | 4 |
| 1 | 0 |  |  |  |  |

Таблиця 10. Варіант порту, для занесення даних

|  |  |  |
| --- | --- | --- |
|  |  | **Порт** |
| 1 | 1 | *P1* |

Таблиця 11. Варіанти часових затримок, що відповідають управляючим сигналам

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Часові затримки** | | | | |
| **Управляючі сигнали** | | | | |
|  |  |  |  |  |
| 1 | 0 | 0 | 18 | 280 | 80 | 720 | 60 |

2. На основі отриманих варіантів виконаємо побудову програми реалізації часових затримок за допомогою асемблеру мікроконтролера MK51.

**Лістинг програми**

;-----Labwork #2------

;Initializing

MOV PSW, #00h ;SEL RB0

MOV P1, 00000000b ;X1 = P1[6] = 0, X2 = P1[7] = 0

block1:

;block 1: y1, y2, y3| 18 mks, 280 mks, 80 mks = 9 \* 2 + 31 \* 2 + 192 /32 + 4 \* 2

MOV R7, #09h ;9 \* 2 = 18 mks

MOV R6, #1Fh ;31 \* 2 = 62 mks

MOV R5, #04h ;4 \* 2 = 8 mks

MOV TH0, #0FAh ; 192/32 = 6 = (-6)DK

MOV TMOD, #00h ; regime 0, using T/L0 timer

ORL P1, 00000111b ;Y1, Y2, Y3 ON

block11: DJNZ R7, block11

ANL P1, 11000110b ;Y1 OFF

block12: DJNZ R6, block12

ANL P1, 11000010b ;Y3 OFF

SETB TCON.4 ;TR0 = 1

block13: JNB TF0, block13

ANL P1, 11000000b ;Y2 OFF

block2:

;X1, checking value of P1[6]

JB P1.6, block5

block3:

;y2| 280 mks = 256/32 + 12 \* 2

MOV R7, #0Ch ;12\*2 = 24 mks

MOV TH0, #0F8h ;256/32 = 8 = (-8)DK

MOV TMOD, #00h ; regime 0, using T/L0 timer

ORL P1, 00000010b ;Y2 ON

SETB TCON.4 ;TR0 = 1

block31: JNB TF0, block31

block32: DJNZ R7, block32

ANL P2, 11000000b; Y2 OFF

block4:

;X2, checking value of P1[7]

JB P1.7, block6

JMP block1 ;return to first block

block5:

;X1, checking value of P1[6]

JB P1.6, block7

block6:

;y1,y4| 18 mks, 720 mks = 9 \* 2 + 704 / 32

MOV R7, #09h ;9\*2 = 18 mks

MOV TH0, #0EAh ;704/32 = 22 = (-22) DK

MOV TMOD, #00h ;0 regime, using T/L0 timer

ORL P1, 00001001b ;Y1,Y4 ON

block61: DJNZ R7, block61

ANL P1, 11001000b ;Y1 OFF

SETB TCON.4 ;TR0 = 1

block62: JNB TF0, block62

ANL P1, 11000000b ;Y4 OFF

block7:

;y4: 720 mks = 704 / 32 + 8 \* 2

MOV R7, #08h ;8 \* 2 = 16 mks

MOV TH0, #0EAh ;704/32 = 22 = (-22)DK

MOV TMOD, #00h ;0 regime, using T/L0 timer

ORL P1, 00001000b ;Y4 ON

SETB TCON.4 ;TR0 = 1

block71: JNB TF0, block71

block72: DJNZ R7, block72

ANL P1, 11000000b ;Y4 OFF

endblock:

END

**Висновок**

В даній лабораторній роботі було розглянуто базову архітектуру і систему команд мікроконтролера МК51. Було виявлено, що даний МК має досить схожу структуру у порівнянні із МК48, але він отримав деякі покращення. Так, тепер всі порти вводу\виводу стали рівноправними, збільшилася кількість лічильників\таймерів, стек отримав окрему структуру, пам'ять даних збільшилася вдвічі(128 б), пам'ять програм в 4 рази (4 кб), збільшилася кількість банків регістрів, а також блок переривань і послідовного порту. Також змінилася і система команд, що дозволяє тепер виконувати всі базові арифметичні операції(включаючи множення і ділення), збільшилася можливість управляючих команд для побудови більш гнучких програм із передачею управління. З’явилася можливість приєднувати до мікроконтролеру додаткову пам'ять програм і даних більшого розміру (до 64кб пам’яті включно). В якості прикладу, було побудовано структурну схему підключення додаткових мікросхем пам’яті даних по 256 б на 10 сторінок, і пам’яті програм по 2 кб на 4 сторінки.

Для прикладу реалізації системи команд було побудовано програму передачі даних із зовнішньої пам’яті у резидентну пам’ять, а також розрахунку деяких чисел за заданим алгоритмом. Побудована програма показує принцип роботи даного алгоритму, а розрахований алгоритм показує приклад його роботи

Також, було розглянуто основні засоби утворення часових затримок при роботі із мікроконтролером МК51. Було виявлено два основних методи побудови затримок: малої і великої тривалості. Для реалізації *перших видів затримок* використовується цикл із декрементом на одному із регістрів мікроконтролера. За рахунок команди переходу на мітку із декрементом, витрачається 2 мкс часу, що дозволяє реалізувати невеликі затримки у часі. Для реалізації *других видів затримок* застосовується 16 розрядні таймери мікроконтролеру, які можуть працювати у 4 режимах, в залежності від значення подільника частоти формувати відповідні затримки за кожну зміну значення таймеру. Так в режимі 0 при f = 12 МГц зміна відбувається через 32 мкс.