**4.1 ИССЛЕДОВАТЬ ВЫЧИСЛЕНИЯ АРИФМЕТИЧЕСКИХ ВЫ-РАЖЕНИЙ**

**Вариант 1.** Рассчитать значение функции Y = 15х+10 ( х изменяется в интервале от 5 до 20 с шагом 1). Результат разместить в РПД с адреса 40h (в массив последовательно занести сначала младший, а затем старший байт ре-зультата).

Mov r0, #40h; начальный адрес массива

Mov r2, #5h, x=5;

M1: Mov a,#15

Mov b, r2

Mul ab

Add a, #10

Mov @r0,a

Inc r0

Mov a,b

Addc a, #0

Mov @r0,a

Inc r0

Inc r2

Cjne r2,#21,ml

Nop

**Вариант 2.** Рассчитать значение функции Y = 3Х+15 ( х изменяется в интервале от 10 до 100 с шагом 10). Результат разместить в РПД с адреса 30h (в массив последовательно занести сначала младший, а затем старший байт результата).

Mov r0,#30h

Mov r1,#38h

Mov r2,#4

Clr c

M1: mov a,@r0

Subb a,@r1

Mov @r0,a

Inc r0

Inc r1

Djnz r2,m1

Nop

**Вариант 3.** Рассчитать значение функции Y = 5Х - 50 ( х изменяется в интервале от 0 до 20 с шагом 2). Результат разместить в РПД с адреса 30h.

Mov r0,#30h

Mov r2,#0

M1: Mov a,#5

Mov b,r2

Mul ab

Clr c

Subb a,#50

Mov @r0,a

Inc r0

Inc r2

Inc r2

Cjne r2,#22,M1

Nop

**Вариант 4.** Составить программу вычитания четырехбайтовых беззна-ковых чисел. Первое число находится в РПД по адресу 30÷33h, второе – по адресу 38÷3Bh. Результат поместить на место первого операнда.

Mov r0,#30h;

Mov r1,#38h

Mov r2,#4

Clr c

M1: Mov a,r0

Subb a,#r1

Mov @r0,a

Inc r0

Inc r1

Djnz r2,M1

nop

**Вариант 5.** Массив чисел был архивирован и помещен в новый массив, в котором предыдущий элемент указывает число, а последующий – количе-ство повторений этого числа в исходном массиве. В результирующем масси-ве описано 8 пар чисел. Найти сумму членов исходного массива. Результат разместить в регистрах R3, R4, R5.

Mov R0,#40h

Mov R2,#8

Mov R3,#0

Mov R4,#0

Mov R5,#0

M1: Mov A,@R0

INC R0

Mov B,@R0

INC RO

MUL AB

ADD A,R3

MOV R3,A

MOV A,R4

ADDC A,B

MOV R4,A

JNC M2

INC R5

M2: DJNZ R2,M1

NOP

**Вариант 6.** Рассчитать 16 значений функции Y=250/Х (для Х, начина-ющегося с 10 с шагом 8). Результаты округлить до целого значения и разме-стить в РПД с адреса 40h.

**Вариант 7.** Перевести однобайтовый шестнадцатеричный операнд в двоично-десятичный упакованный формат. Исходный операнд находится в регистре R5. Результат разместить в регистрах R4 (число сотен) и R3 (десят-ки, единицы).

mov A, R5 ; Пересылка операнда в Акк.

mov B, #100 ; Делитель 100 → B

div AB ; Акк. содержит число сотен

mov R4, A ; Пересылка числа сотен → R4

mov A, B ; Пересылка остатка → Акк.

mov B, #10 ; Делитель 10 → B

div AB ; Десятки → Акк., единицы → B

swap A ; Пересылка десятков в старшую тетраду

add A, B ; Пересылка единиц → Акк.

mov R3, A ; Пересылка результата → R3

**Вариант 8.** В РПД, начиная с адреса 30h находится массив из 20 эле-ментов. Подсчитать количество элементов массива, попавших в интервал от 50 до 100. Результат запомнить в регистре R5.

**Вариант 9.** В РПД, начиная с адреса 30Н, находится массив из 16 чи-сел. Найти максимальный элемент массива и поместить в R2 его значение, а в R3 его адрес. 46

Mov ro,#30h; начальный адрес массива

Mov r4,#15; счетчик повторения

Mov 2,@r0

Mov 3,ro

M1: Inc r0

Mov a,@r0

Cjne a,2,M2

M2: Jnc M3

Mov 2,@r0

Mov 3,ro

M3: djnz r4,M1

Nop

**Вариант 10.** В регистре R5 находится двоично-десятичный операнд. Перевести операнд в шестнадцатеричное значение и поместить в R5

**4.3 ИССЛЕДОВАТЬ ПРИМЕНЕНИЕ КОМАНДЫ CJNE**.

**Вариант 1.** В РПД, начиная с адреса 20h, находится массив из 16 эле-ментов. Подсчитать и сохранить в регистрах: R2 – количество элементов массива, меньших значения 128; R3 – количество элементов массива, равных 128; R4 – количество элементов массива, больших 128.

MOV R0,#20h

MOV R1,#16

MOV R2,#0

MOV R3,#0

MOV R4,#0

M1: CJNE @R0,#128,M2

INC R3

JMP M4

M2: JC M3; Переход, если меньше 128

INC R4

JMP M4

M3: INC R2

M4: INC R0

DJNZ R1,M1

NOP

**Вариант 2.** В РПД с адреса 20h находится массив, состоящий из 16 элементов. Суммировать элементы массива до тех пор, пока значение суммы не превысит 512. Выдать в R3 номер элемента, на котором произошло пере-полнение. Если сумма элементов не достигла значения 512, то выдать в реги-стре R3 значение 0.

**Вариант 3.** Для функции Y=20Х+45 выдать в R2 первое значение ар-гумента, при котором значение функции превысит 1024. Начальное значение аргумента Х=10.

Mov r4,#0

M1: Inc r4

Mov a,#20

Mov b,r4

Mul ab

Add a,#45

Mov a,b

Addc a,#0

Cjne a,#4,M1

Nop

**Вариант 4.** В РПД с адреса 20h находится массив из 16 чисел. Элемен-тами массива являются числа 32, 64, 96 и 128. Подсчитать и сохранить в ре-гистрах R4 ÷ R7 количество повторений каждого элемента. 47

Mov r0, #20h

Mov r2, #16

Mov r4, #0

Mov r5, #0

Mov r6, #0

Mov r7, #0

M1: cjne @r0, #32, m2

Inc r4

Jmp con

M2: cjne @r0, #64, m3

Inc r5

Jmp con

M3: cjne @r0, #96, m4

Inc r6

Jmp con

M4: inc r7

Con: inc r0

Djnz r2, m1

nop

**Вариант 5.** В РПД по адресам 20h÷2Fh находится массив. С адреса 30h создать массив, в который входят адреса элементов первого массива, равных 128. В регистре R2 сохранить число элементов, равных 128. Прервать вы-полнение программы, если будет найдено 5 элементов со значением 128.

mov r0,#20h

mov r1,#30h

mov r2,#0

m1: cjne @r0,#128,m2

inc r2

mov @r1,0

inc r1

cjne r2,#5,m2

jmp con

m2: inc r0

cjne r0,#30h,m1

con: nop

**Вариант 6.** В РПД с адресов 20h и 30h находятся 2 массива, состоящие из 16 элементов каждый. Подсчитать количество элементов первого массива, которые имеют равные значения во 2 массиве. Результат занести в регистр R2.

Mov r0, #20h

Mov r3, #16

Mov r2, #0

M1: Mov a, @r0

Mov r1, #30h

Mov r4, #16

M2: Mov b, @r1

Cjne a, b, m3

Inc r2

Jmp m4

M3: Inc r1

Djnz r4, m2

M4: Inc r0

Djnz r3, m1

Nop

**Вариант 7.** Для функции Y=40Х+10 получить первое значение, пре-вышающее 512, начиная с Х=1. Значение аргумента записать в R4, функции - в R5, R6.

Mov r4,#0

M1: Inc r4

Mov a,#40

Mov b,r4

Mul ab

Add a,#10

Mov r5,a

Mov a,b

Addc a,#0

Mov r6,a

Cjne a,#2,M1

Nop

**Вариант 8.** В ВПД, начиная с адреса 100h, находится массив из 10 элементов. Получить в регистре R3 число элементов, равных 55h. Счет пре-рвать, если число элементов превысит 3.

**Вариант 9.** Для функции 15Х+85 найти первое значение аргумента, при котором младший байт функции равен 155.

Mov r2,#0;

M1: inc r2

Mov a,#15

Mov b,r2

Mul ab

Add a,#85

Cjne a,#155,m1

nop

**Вариант 10.** В ВПД с адреса 300h находится массив из 15 чисел. Эле-ментами массива являются числа 10, 20, 30 и 180. Подсчитать и сохранить в регистрах R4 - R7 количество повторений каждого элемента.

**4.4 ИССЛЕДОВАТЬ ПРИМЕНЕНИЕ ТАБЛИЧНЫХ ДАННЫХ** при программировании микроконтроллера.

**Вариант 1.** Создать в ВПД, начиная с адреса 40h, массив из 10 чисел, элементами которого являются квадраты чисел (от 0 до 15), прочитанных из порта P1. Таблица квадратов чисел расположена в ПЗУ, начиная с адреса 100h.

MOV R0,#10  
MOV DPTR,[#100h](tg://search_hashtag?hashtag=100h)  
MOV R1,[#40h](tg://search_hashtag?hashtag=40h)

M2: MOV A,P1  
MOVC A, @A+DPTR  
MOVX @R1,A  
INC R1  
DJNZ R0,M2  
NOP  
ORG 100h  
DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225

**Вариант 2.** Для кодирования информации используется алгоритм пе-рестановки, суть которого состоит в том, что символы исходного алфавита заменяются на те же символы, но в другом порядке. Соответствие символов исходного алфавита – кодированному приведено в таблице 4.1. Для опреде-ления конца сообщения используется символ «\*» с кодом 2Аh.

В РПД с адреса 30h находится сообщение неизвестной длины (не более 20 символов). Закодировать сообщение в соответствии с таблицей 4.1 и раз-местить в РПД с адреса 50h. 48



Mov R0, #30h

Mov R1, #50h

Mov dptr, #0

M1: mov A, @R0

Cjne A, #2Ah, M2

Jmp con

M2: movc A, @A+dptr

Mov @R1, A

Inc R0

Inc R1

Jmp M1

Con: nop

Org 5Fh

Db 0d4h

Org 0c0h

Db 0c7h, 0ceh, 0d5h, …

Db 0c2h, 0d0h, 0d7h, …

Db 0c4h, 0d2h, 0d9h, …

**Вариант 3.** Для кодирования информации используется алгоритм пе-рестановки, суть которого состоит в том, что символы исходного алфавита заменяются на те же символы, но в другом порядке. Соответствие символов исходного алфавита – кодированному приведено в таблице 4.1. Для опреде-ления конца сообщения используется символ «\*» с кодом 2Аh.

В РПД с адреса 20h находится закодированное сообщение длиной 29 символов. Текст сообщения:

«ГЭЭОМИЧЦИФКЗГЙЭЯЦКЭХЗЧЭФХИКЧЭ\*».

Mov r0,#20h

Mov r1,#50h

Mov dtpr,#0

M1: cjne @r0,#2ah,M2

Jmp con

M2: mov b,@r0

Mov dpl,#5eh

M3: inc dpl

Clr a

Movc a,@a+dptr

Cjne a,b,M3

Mov @r1,dpl

Inc r0

Inc r1

Jmp M1

Con: NOP

Org 5fh

Db 0d4h

Org 0c0h

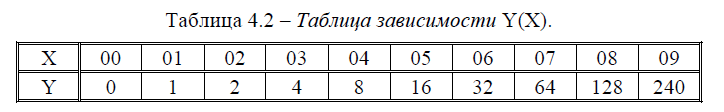
Db 0c7h,0ceh,0d5h,0dbh,0dfh,0c8h,0c0h,0cfh,0d6h,0dch,0c9h

Db 0c2h,0d0h,0d7h,0ddh,05fh,0cah,0c3h,0d1h,0d8h,0deh,0cbh

Db 0c4h,0d2h,0d9h,0cch,0c5h,0cdh,0d3h,0dah,0c6h,0c1h

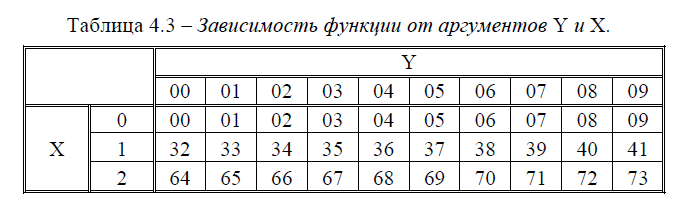
Раскодировать сообщение в соответствии с таблицей 4.1 и разместить текст раскодированного сообщения в РПД с адреса 50h.

**Вариант 4.** Зависимость Y(X) представлена в таблице 4.2 (значения заданы в десятичном виде).



Составить программу нахождения аргумента по значению функции. Значение функции (Y) находится в регистре R7. Значение аргумента (X) раз-местить в регистре R0. Если значение (Y) отсутствует в таблице, то выдать в регистре R0 значение 0FFh.

**Вариант 5.** Зависимость функции от двух аргументов X и Y задана в табличном виде (таблица 4.3). Значения в таблице даны в десятичном виде.



Cоставить программу для выдачи в порт Р2 значений функции в зави-симости от аргументов, которыми являются входные сигналы портов Р0 (значение Х) и Р1 (значение Y).

MOV DPTR,#TAB

MOV A,P0

MOV B,#10

MUL AB

ADD A,P1

MOVC A,@A+DPTR

MOV P2,A

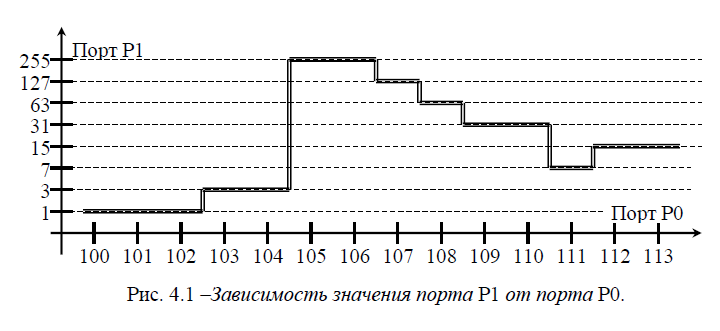
NOP

TAB: DB 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

DB 32, 33, 34, 35, 36, 37, 38, 39, 40, 41

DB 64, 65, 66, 67, 68, 69, 70, 71, 72, 73

**Вариант 6.** В зависимости от осведомительных сигналов, поступаю-щих в порт Р0, в порт Р1 выдаются управляющие сигналы. При значении порта Р0, меньшем 100, в порт Р1 выдается 00, а при значении порта Р0, большем 113 – 255. Зависимость выходного значения порта Р1 от входного значения порта Р0 в интервале 100-113 определяется графиком, представ-ленным на рис. 4.1.



Mov dptr, #tab

Mov a, p0

Cjne a, #100, M1

M1: Jnc M2

Mov p1, #0

Jmp con

M2: cjne a, #114, M3

M3: jc M4

Mov p1, #255

Jmp con

M4: clr c

Subb a, #100

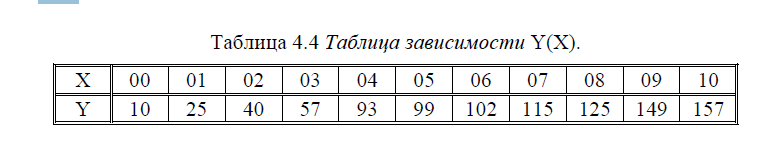
Movc a, @a+dptr

Mov p1, a

Con: nop

Tab: db 1, 1, 1, 3, 3, 255, 255, 127, 63, 31, 31, 7, 15, 15

**Вариант 7.** Зависимость значения функции от значения аргумента представлена в таблице 4.4 в десятичном виде. Составить программу, выда-ющую в порт Р2 значение Х в зависимости от поступившего в порт Р0 значе-ния Y. Если значение Y отсутствует в таблице, то выдать в порт Р0 значение FFh.



mov dptr,#tab

mov r0,#0

M1: mov a,r0

movc      a,@a+dptr

cjne a,P0,M2

mov P2,r0

Jmp con

M2: inc r0

cjne r0,#11,M1

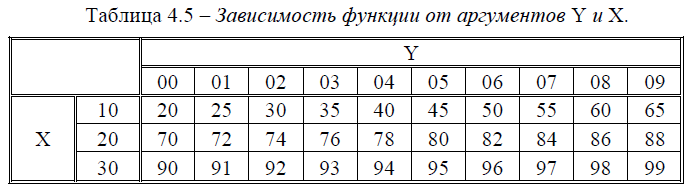
mov P2,#0FFh

Con: nop

Tab: db 10, 25, 40, 57, 93, 99, 102, 115, 125, 149, 157

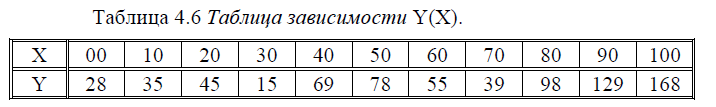
**Вариант 8.** Зависимость функции от двух аргументов Х и Y представ-лена в табличном виде (таблица 4.5). Значения даны в десятичной системе счисления.

Составить программу, которая в зависимости от значения функции, поступившего в порт Р0, выдает в порты Р1 и Р2 значения X и Y.



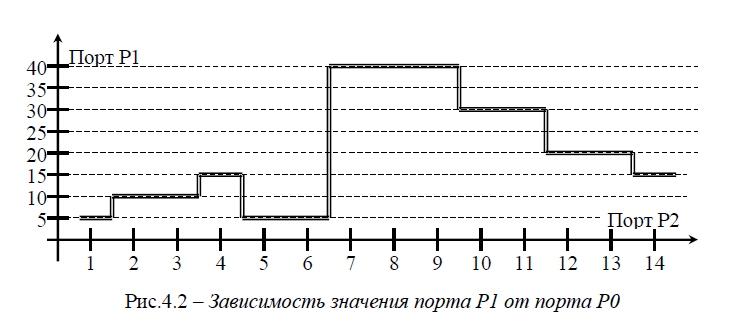
**Вариант 9.** Зависимость значения функции от значения аргумента представлены в таблице 4.6 в десятичном виде.

В ВПД, начиная с адреса 10h, располагается массив из 10 элементов. Разместить с адреса 20h массив, элементы которого являются функциями от элементов первого массива в соответствии с таблицей 4.6. Если значение X отсутствует в таблице, записать вместо Y – значение 255.



**Вариант 10.** Составить программу, выдающую в порт Р1 значение в зависимости от входного значения порта Р2.

Зависимость выходных значений от входных представлена графиком (рис.4.2) в десятичной системе счисления. При превышении допустимого входного значения в порт выдается значение 00.



**4.5 ИССЛЕДОВАТЬ ПРИМЕНЕНИЕ ПОДПРОГРАММ**. В соот-ветствии с вариантом разработать программу с использованием под-программы.

**Вариант 1.** В порты микроконтроллера Р0-Р3 поступают двоично-десятичные данные. Перевести данные в шестнадцатеричный формат и раз-местить в РПД последовательно с адреса 30h.

MOV R0,#30h

MOV R5,P0

CALL Fish

MOV R5,P1

CALL Fish

MOV R5,P2

CALL Fish

MOV R5,P3

CALL Fish

NOP

Fish: MOV A, R5 ; Пересылка операнда → Акк.

ANL A, #0Fh ; Выделение младшей тетрады

XCH A, R5 ; Младшую тетраду → R5

ANL A, #0F0h ; Выделение старшей тетрады

SWAP A ; Перестановка тетрад

MOV B, #10 ; Множитель 10 → B

MUL AB ; Десятки исходного числа → А

ADD A, R5 ; Результат → Акк.

MOV @R0, A

INC R0

RET

**Вариант 2.** Выдать последовательно в порты Р1 и Р2 микроконтролле-ра содержимое младших байт счетчиков в двоично-десятичном формате (в Р1 – сотни, в Р2 – десятки и единицы).

**Вариант 3.** В порты Р0÷Р3 поступают шестнадцатеричные данные. За-нести в РПД, начиная с адреса 40h количество единиц, поступивших в каж-дый порт.

Mov r0,#40h

Mov a,p0

Call cat

Mov a,p1

Call cat

Mov a,p2

Call cat

Mov a,p3

Call cat

NOP

Cat: mov r2,#0

Mov r3,#8

M1: RRC a

Jnc M2

Inc r2

M2: djnz r3,M1

Mov @r0,2

Inc r0

Ret

**Вариант 4.** Для каждого из регистров R0, R3 и регистра-расширителя В последовательно выдать в порты информацию о содержимом регистров:

* Р0 - прямое значение байта;
* Р1 - инверсное значение байта;
* P2 - количество нолей в байте;
* Р3.0 - флаг контроля на четность.

Mov a,r0

Call Sasha

Mov a,r3

Call Sasha

Mov a,b

Call Sasha

Nop

Sasha: mov p0,a

Mov c,p

Mov p3.0,c

Cpl a

Mov p1,a

Mov r2,#0

Mov r4,#8

M1: rrc a

Jnc m2

Inc r2

M2: djnz r4,m1

Mov p2,r2

ret

**Вариант 5.** В каждый из портов Р0÷Р2 поступают данные от двух че-тырехразрядных датчиков. Выдать в порт Р3 сумму шести датчиков, под-ключенных к портам Р0÷Р2.

Mov R2,#0

Mov a,p0

Call dev

Mov a,p1

Call dev

Mov a,p2

Call dev

Mov p3,r2

Nop

Dev: Mov b,a ;копировать

Anl b,#0Fh ;Выдиление младшей тетрады

Anl a,#0f0H ;Выделение старшей тетрады

Swap a

Add a,b

Add a,r2

MoV r2,a

Ret

**Вариант 6.** Записать в регистры R3, R7 и регистр-расширитель В про-изведение их старшей и младшей тетрады соответственно.

**Вариант 7.** Выдать в порты Р0÷Р2 количество единиц, содержащихся в регистрах R0,R7 и регистре-расширителе В соответственно.

Mov a,r0

Call fish

Mov p0,r2

Mov a,r7

Call fish

Mov p1,r2

Mov a,b

Call fish

Mov p2,r2

Nop

fish: Mov r2,#0

Mov r4,#8

M1: Rrc a

Jnc M2

Inc r2

M2: djnz r4,M1

Ret

**Вариант 8.** Считать с интервалом в 10 милисекунд 5 значений из порта Р0, используя подпрограмму задержки на 10 милисекунд. Входные коды за-писать в РПД, начиная с адреса 40h

**Вариант 9.** На вход внешнего прерывания ~INT0 микроконтроллера через схему «8ИЛИ-НЕ» подключено 8 источников прерываний ИСТ0- ИСТ7. Эти же источники подключены к выводам порта Р2. Составить под-программу обслуживания прерывания, которая определяет номер источника прерывания и переходит по соответствующему адресу. В cлучае одновре-менного поступления нескольких запросов вступает в действие система при-оритетов, которая имеет вид в порядке убывания: ИСТ3, ИСТ5, ИСТ7, ИСТ4, ИСТ0, ИСТ1, ИСТ2, ИСТ4. Адреса начала подпрограмм установить самосто-ятельно.

**Вариант 10.** Составить подпрограмму, которая по сигналу внешнего прерывания ~INT1 выдает в порты Р0÷Р2 текущую сумму таймеров-счетчиков.