Х = Х3\*Х4 + Х7/Х8 + Х9\*Х9 + Х10\*Х10

Ниже представлен алгоритм выполнения функции Х

Х3\*Х4

R1 - старшие разряды Х3

R2 - младшие разряды Х3

R3 - старшие разряды Х4

R4 - младшие разряды Х4

R6=0 R5 = 0

R7 = 10H - счётчик

R0= 10H - адрес начала

Результат: в памяти по адресам

13h 12h 11h 10h

P3:=L1(P3)

P2:=R1(P2)

C:= C - 1

C := 16

C = 0

Х1\*=1

P2 := P2 + P1

Нет

Да

Да

Нет

;Основная программа данные считываем с порта Р1 используем банк регистра нулевой

SEL RBO

;========Считываем с порта Р1 значение Х3============

IN A,P1

MOV R1,A ; старшие разряды Х3

IN A,P1

MOV R2,A; младшие разряды Х3

;========Считываем с порта Р1 значение Х3============

IN A,P1

MOV R3,A ; старшие разряды Х4

IN A,P1

MOV R4,A; младшие разряды Х4

;========Производим подготовку к умножению Х3\*Х4===

MOV R6,#0

MOV R5,#0

MOV R7,#10H ; счётчик в умножении

MOV R0,#10H ; адрес начала младших разрядов результата

CALL MULL; Вызов подпрограммы умножения

|  |  |  |  |
| --- | --- | --- | --- |
| 13h | 12h | 11h | 10h |

;результат получили Х3\*Х4=

;========Считываем с порта Р1 значение Х7============

IN A,P1

MOV R2,A ; старшие разряды Х7

IN A,P1

MOV R3,A; младшие разряды Х7

;========Считываем с порта Р1 значение Х8============

IN A,P1

MOV R4,A ; старшие разряды Х8

IN A,P1

MOV R5,A; младшие разряды Х8

;========Производим подготовку к делению Х7/Х8===

MOV R6,#0

MOV R7,#0

CALL DIVI ; вызов подрограммы деления

; результат получаем в **R6.R7**

; записуем результат в память

MOV RO,#14H

MOV A,R6

MOV @RO,A ; запись частного в память

MOV RO,#9H

MOV A,R7

MOV @RO,A ; запись остатка в память

;========Считываем с порта Р1 значение Х9============

IN A,P1

MOV R1,A ; старшие разряды Х9

MOV R3,A ; старшие разряды Х9

IN A,P1

MOV R2,A ; младшие разряды Х9

MOV R4,A ; младшие разряды Х9

;========Производим подготовку к умножению Х9\*Х9===

MOV R6,#0

MOV R5,#0

MOV R7,#10H ; счётчик в умножении

MOV R0,#15H ; адрес начала младших разрядов результата

CALL MULL; Вызов подпрограммы умножения

|  |  |  |  |
| --- | --- | --- | --- |
| 18h | 17h | 16h | 15h |

; результат получили Х9\*Х9=

;========Считываем с порта Р1 значение Х10============

IN A,P1

MOV R1,A ; старшие разряды Х10

MOV R3,A ; старшие разряды Х10

IN A,P1

MOV R2,A ; младшие разряды Х10

MOV R4,A ; младшие разряды Х10

;========Производим подготовку к умножению Х10\*Х10===

MOV R6,#0

MOV R5,#0

MOV R7,#10H ; счётчик в умножении

MOV R0,#19H ; адрес начала младших разрядов результата

CALL MULL; Вызов подпрограммы умножения

|  |  |  |  |
| --- | --- | --- | --- |
| 22h | 21h | 20h | 19h |

; результат получили Х9\*Х9=

;========Производим суммирование результатов===

; Х = Х3\*Х4 + Х7/Х8 + Х9\*Х9 + Х10\*Х10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Х3\*Х4 | 13H | 12H | 11H | 10H |
| Х7/Х8 |  |  |  | 14H . 9H |
| Х9\*Х9 | 18H | 17H | 16H | 15H |
| Х10\*Х1 | 22H | 21H | 20H | 19H |
| X | 13H | 12H | 11H | 10H . 9H |

;========Производим суммирование двух первых операндов Х3\*Х4 + Х7/Х8 ===

MOV RO,#10H

MOV A,@RO ; считывание младших разрядов результата умножения Х3\*Х4

MOV R2,A

MOV R1,#14H

MOV A,@R1 ; считывание частного от деления Х7/Х8

ADD A,R2 ;суммирование результата и фиксация признака переноса

MOV @R0,A ; запись результата в память в ячейку 10h

MOV RO,#11H

MOV A,@RO ; считывание разрядов результата умножения Х3\*Х4

ADDС A,#0 ; суммирование результата и переноса

MOV @R0,A ; запись результата в память в ячейку 11h

MOV RO,#12H

MOV A,@RO ; считывание разрядов результата умножения Х3\*Х4

ADDС A,#0 ; суммирование результата и переноса

MOV @R0,A ; запись результата в память в ячейку 12h

MOV RO,#13H

MOV A,@RO ; считывание разрядов результата умножения Х3\*Х4

ADDС A,#0 ; суммирование результата и переноса

MOV @R0,A ; запись результата в память в ячейку 13h

;========Производим Суммирование двух первых операндов (Х3\*Х4 + Х7/Х8) и Х9\*Х9 ===

CLR C

MOV RO,#16H

MOV A,@RO ; считывания разрядов результата умножения Х9\*Х9

MOV R2,A

MOV R1,#10H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8)

ADD A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 10h

MOV RO,#17H

MOV A,@RO ; считывания разрядов результата умножения Х9\*Х9

MOV R2,A

MOV R1,#11H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 11h

MOV RO,#18H

MOV A,@RO ; считывания разрядов результата умножения Х9\*Х9

MOV R2,A

MOV R1,#12H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 12h

MOV RO,#19H

MOV A,@RO ; считывания разрядов результата умножения Х9\*Х9

MOV R2,A

MOV R1,#13H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 13h

;========Производим Суммирование трёх первых операндов (Х3\*Х4 + Х7/Х8 + Х9\*Х9) и Х10\*Х10 ===

CLR C

MOV RO,#19H

MOV A,@RO ; считывания разрядов результата умножения Х10\*Х10

MOV R2,A

MOV R1,#10H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8 + Х9\*Х9)

ADD A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 10h

MOV RO,#20H

MOV A,@RO ; считывания разрядов результата умножения Х10\*Х10

MOV R2,A

MOV R1,#11H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8 + Х9\*Х9)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 11h

MOV RO,#21H

MOV A,@RO ; считывания разрядов результата умножения Х10\*Х10

MOV R2,A

MOV R1,#12H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8 + Х9\*Х9)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 12h

MOV RO,#22H

MOV A,@RO ; считывания разрядов результата умножения Х10\*Х10

MOV R2,A

MOV R1,#13H

MOV A,@R1 ; считывание разрядов результата cуммирования (Х3\*Х4 + Х7/Х8 + Х9\*Х9)

ADDC A,R2 ; суммирование результататов и фиксация признака переноса

MOV @R1,A ; запись результата в память в ячейку 13h

;======== В результате получили : ==================

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | 13H | 12H | 11H | 10H . 9H |

;======== ПОДПРОГРАММ УМНОЖЕНИЯ ДВОХ ЧИСЕЛ Х\*У =======

; R0 – адрес начала млаших разрядов результата

; R1 – старшие разряды Х

; R2 – младшие разряды Х

; R3 – старшие разряды У

; R4 – младшие разряды У

; R5 – дополнительный регистр

; R6 – дополнительный регистр

; R7 – счётчик

MUL:

LL0:

MOV A,R1

JB7 LL1 ; Проверка Х1\* (старшего разряда)

LL3: ; X1\*=0;

; P1 сдвигаем вправо ===================================

MOV A,R3 ; Старшиер разряды в А

CLR C

RRC A ; Сдвиг А вправо А[7] = 0 фиксируем признак **с**

MOV R3,A

MOV A,R4 ; разряды в А

RRC A ; Сдвиг А вправо А[7] = **с** фиксируем признак **с**

MOV R4,A

MOV A,R5 ; разряды в А

RRC A ; Сдвиг А вправо А[7] = **с** фиксируем признак **с**

MOV R5,A

MOV A,R6 ; Младшие разряды в А

RRC A ; Сдвиг А вправо А[7] = **с** фиксируем признак **с**

MOV R6,A

CLR C ; очистка признак **с**

; P3 сдвигаем вправо ===================================

MOV A,R2

RLC A ; Сдвиг А вправо А[0] = 0 фиксируем признак **с**

MOV R2,A

MOV A,R1

RLC A ; Сдвиг А вправо А[0] = **с** фиксируем признак **с**

MOV R1,A

JMP CHECK ; Прыжок на метку

LL1:

; Х1\* = 1 ========================================================

; Р2 = Р2 + Р1

; Считываем данные и сумируем. Резльтат записываем в память

MOV A,@R0 ; считали данные

ADD A,R6 ; просумировали

MOV @RO,A ; результат обратно в память

INC R0 ; R0 = R0 + 1 - для получения адреса следующих разрядов

MOV A,@R0 ; считали данные

ADDС A,R5 ; просумировали с учётом переноса

MOV @RO,A ; результат обратно в память

INC R0 ; R0 = R0 + 1 - для получения адреса следующих разрядов

MOV A,@R0 ; считали данные

ADDС A,R4 ; просумировали с учётом переноса

MOV @RO,A ; результат обратно в память

INC R0 ; R0 = R0 + 1 - для получения адреса следующих разрядов

MOV A,@R0 ; считали данные

ADDС A,R3 ; просумировали с учётом переноса

MOV @RO,A ; результат обратно в память

JMP LL3

CHECK:

DJNZ R7,LLO ; прыжок на метку LLO если R7 не равно нулю и декримент R7

RET ; Возврат из подрограммы

;======== ПОДПРОГРАММ ДЕЛЕНИЯ ДВОХ ЧИСЕЛ Х\*У =======

; R2 – старшие разряды Х

; R3 – младшие разряды Х

; R4 – старшие разряды У

; R5 – младшие разряды У

; R0 – счётчик

DIVI:

MOV R0, #10H

MOV R1,#32H

MOV A,R4

MOV @R1,A ; Пересылка во внутр. Память старших разрядов

MOV R1,#33H

MOV A,R5

MOV @R1,A ; Пересылка во внутр. Память старших разрядов

MOV R4,#0

MOV R5,#0

SUBI:

CLR C

MOV R1,#35H

MOV A,@R1 ;Считывание с памяти разрядов

CPL A ; Инвертирование А

INC A ; Получаем А в доп коде

ADD A,R5

MOV R5,A ;Промежуточные данные в R5

MOV R1,#34H

MOV A,@R1 ;Считывание с памяти разрядов

CPL A ; Инвертирование А

ADDC A,R4

MOV R4,A ;Промежуточные данные в R4

MOV R1,#33H

MOV A,@R1 ;Считывание с памяти разрядов

CPL A ; Инвертирование А

ADDC A,R3

MOV R3,A ;Промежуточные данные в R3

MOV R1,#32H

MOV A,@R1 ;Считывание с памяти разрядов

CPL A ; Инвертирование А

ADDC A,R2

MOV R2,A ;Промежуточные данные в R2

JMP SH

ADDI:

CLR C

MOV R1,#35H

MOV A,@R1 ;Считывание с памяти разрядов

ADD A,R5

MOV R5,A ;Промежуточные данные в R5

MOV R1,#34H

MOV A,@R1 ;Считывание с памяти разрядов

ADDC A,R4

MOV R4,A ;Промежуточные данные в R4

MOV R1,#33H

MOV A,@R1 ;Считывание с памяти разрядов

ADDC A,R3

MOV R3,A ;Промежуточные данные в R3

MOV R1,#32H

MOV A,@R1 ;Считывание с памяти разрядов

ADDC A,R2

MOV R2,A ;Промежуточные данные в R2

JMP SH

SH:

NOP

CLR C

CLR F1

JB7 SHZ

CPL C

CPL F1

SHZ:

MOV A,R7

RLC A

MOV R7,A

MOV A,R6

RLC A

MOV R6,A

DJNZ R0, OK

JMP ENDING

OK:

MOV R1,#35H

MOV A,@R1 ;Считывание с памяти разрядов

RRC A

MOV @R1,A ;Запись обратно в память

MOV R1,#34H

MOV A,@R1 ;Считывание с памяти разрядов

RRC A

MOV @R1,A ;Запись обратно в память

MOV R1,#33H

MOV A,@R1 ;Считывание с памяти разрядов

RRC A

MOV @R1,A ;Запись обратно в память

MOV R1,#32H

MOV A,@R1 ;Считывание с памяти разрядов

RRC A

MOV @R1,A ;Запись обратно в память

CLR C

JF1 SUBI

JMP ADDI

ENDING:

RETR ; Возврат из подрограммы