

2021.02.03

Given a 4 x 4 matrix of WORD (i.e. 16 bits single data) SOURCE write a 8086 assembly program which rotates the rows of SOURCE from up to down by $1 \leq n \leq 3$ positions and stores the result in the matrix DESTINATION, with n given by the user. The choice is yours about how to store the matrices in the memory. Please add significant comments to the code and instructions. If you have time, in order to get one additional point, provide the instructions to extend the program to consider n in the range - $80 \leq n \leq +120$

Example:

Initial matrix SOURCE

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

if n=3 DESTINATION becomes

E	F	G	H
I	J	K	L
M	N	O	P
A	B	C	D

if n=1 DESTINATION becomes

M	N	O	P
A	B	C	D
E	F	G	H
I	J	K	L

2021.02.22

Given a 3 x 3 matrix of bytes SOURCE representing unsigned numbers, write a 8086 assembly program which computes (in circular buffer mode) the addition of each row element with the corresponding same column element in the row immediately below and stores the result on 16 bits in the same position of a matrix DESTINATION. The last row elements do add up with the corresponding first row elements (i.e. circular buffer mode). Please add significant comments to the code and instructions.

Example:

Initial matrix SOURCE

1	2	3
4	5	6
7	8	9

the following matrix DESTINATION is computed

5	7	9
11	13	15
8	10	12

2021.06.23

Given a 5 x 5 matrix of bytes SOURCE representing unsigned numbers, write a 8086 assembly program which computes on 16 bits the sum of all cells excluding these on the main diagonal, i.e. upper left-to-lower-right diagonal, minus the sum of all the cells of the same main diagonal.

Please add significant comments to the code and instructions.

Friendly advice: before starting to write down the code, think at a possible (very) simple algorithm! The choice of the algorithm highly influences the complexity and length of the code.

Example:

matrix SOURCE

1	2	3	4	5
6	7	8	9	0
9	8	7	6	5
4	3	2	1	0
7	7	7	7	7

all cells excluding the main diagonal:

2+3+4+5+

6+8+9+0+

9+8+6+5+

4+3+2+0+

7+7+7+7= 102

all cells on the main diagonal

1+

7+

7+

1+

7= 23

Result (on 16 bits in two's complement) = 102-23 = 79

2021.08.31

Given a 8 x 5 matrix of bytes SOURCE representing unsigned numbers, write a 8086 assembly program which computes on 16 bits (two's complement) the addition of all cells with indexes (i,j) where i+j is an even value, minus all the cells whose i+j is an odd value. Please consider that i ranges from 0 to 7 and j ranges from 0 to 4.

Please add significant comments to the code and instructions.

Friendly advice: before starting to write down the code, think at a possible (very) simple algorithm! The choice of the algorithm highly influences the complexity and length of the code.

Example:

matrix SOURCE

1	2	3	4	5
6	7	8	9	0
9	8	7	6	5
4	3	2	1	0
7	7	7	7	7
3	5	7	9	0
8	7	6	5	4
9	9	9	3	2

the cells with i+j even are added up, while the cells with i+j odd are subtracted

1+3+5+7+9+9+ ...

-2-4-6-8-0-....

The result will be clearly on 16 bits in two's complement.