11.04.2020 СМАРАНДИ

**Л/Р 1 «Команды языка Ассемблер». Вариант 020.**

**Задача 1 (на один плюсик):**

Присвоить в регистр AX какое-либо небольшое начальное значение. Научиться умножать на степени двойки, используя сдвиг:

BX = 4\*AX

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

MOV AX, 1

SHL AX, 2

MOV BX, AX

;---[ Стандартное завершение программы ]----------------------------------------

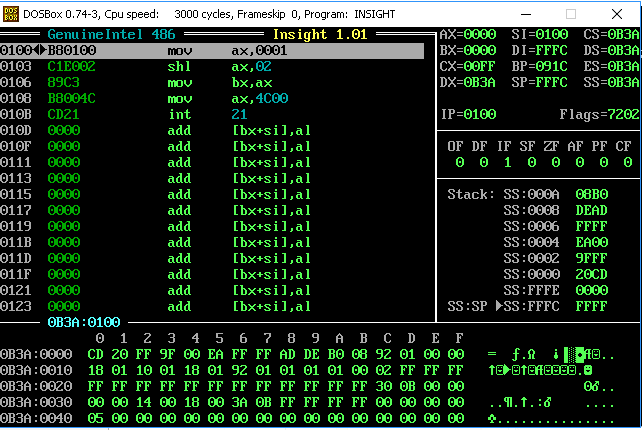
mov AX, 4C00h

int 21h

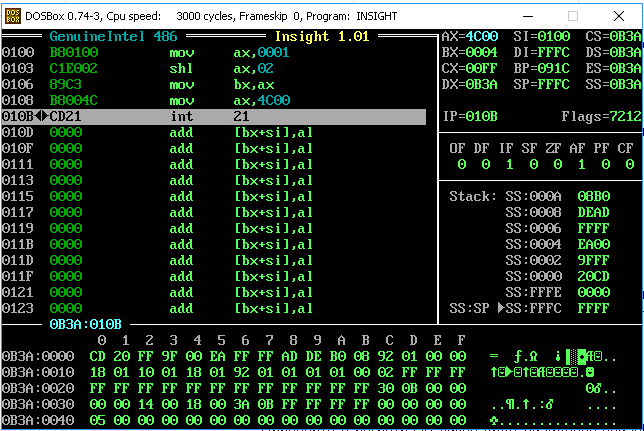
;===[ Начало сегмента данных ]==========================================

; ...

1. скриншот



2. скриншот



**Задача 2 (на один плюсик):**

Присвоить в регистр AX какое-либо небольшое начальное значение. Научиться умножать на произвольные константы, комбинируя сдвиги и сложения:

BX = 9\*AX

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov ax, 1

mov dx, ax

shl ax, 3

mov bx, ax

mov ax, dx

add bx, ax

;---[ Стандартное завершение программы ]----------------------------------------

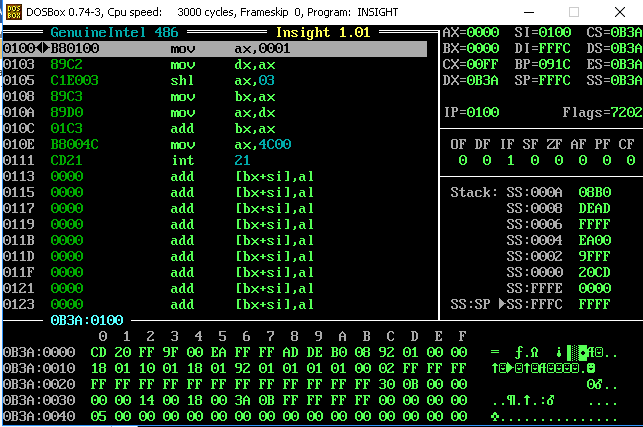
mov AX, 4C00h

int 21h

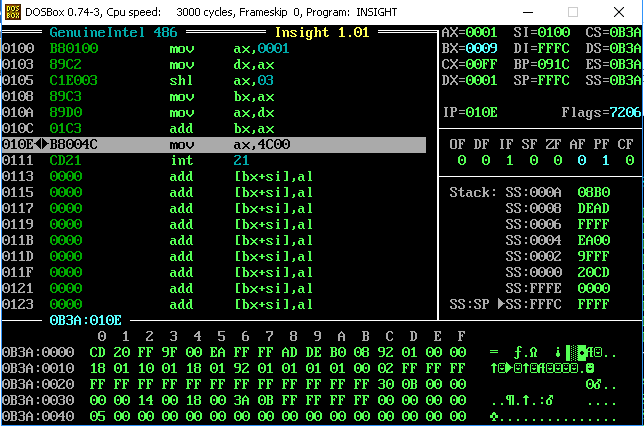
;===[ Начало сегмента данных ]==========================================

; ...

1. скриншот



2. скриншот



**Задача 3 (на один плюсик):**

Присвоить в регистр AX какое-либо начальное значение (например, 12A8h). Научиться выделять нужную цифру числа, используя наложение маски. Маска должна быть универсальной, т.е. правильно работать для любого начального значения AX:

AX = 12A8h → AX = 0008h

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov ax, 12A8h

and ax, 000fh

;---[ Стандартное завершение программы ]----------------------------------------

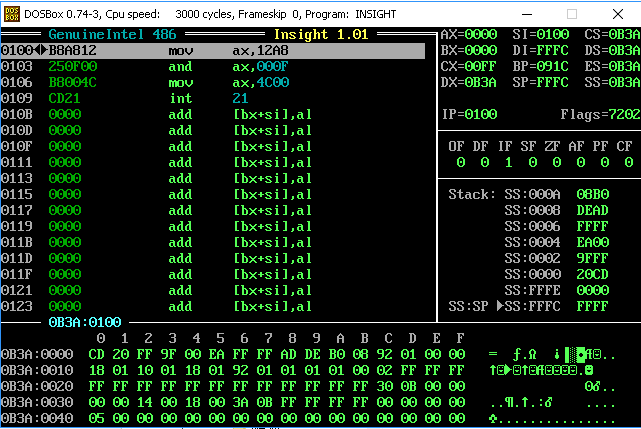
mov AX, 4C00h

int 21h

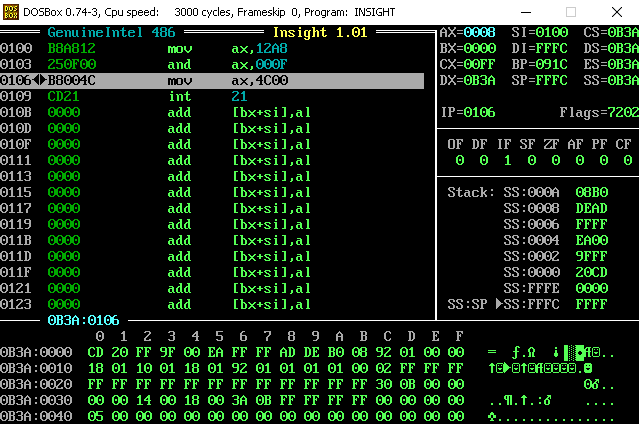
;===[ Начало сегмента данных ]==========================================

; ...

1. скриншот



2. скриншот



**Задача 4 (на один плюсик):**

Присвоить в регистр AX какое-либо начальное значение (например, 12A8h). Научиться выделять нужную цифру числа, используя наложение маски, а также перемещать ее в нужную позицию через сдвиги:

AX = 12A8h → AX = 8000h

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov ax, 12A8h

and ax, 000fh

mov bx, ax

shl bx, 12

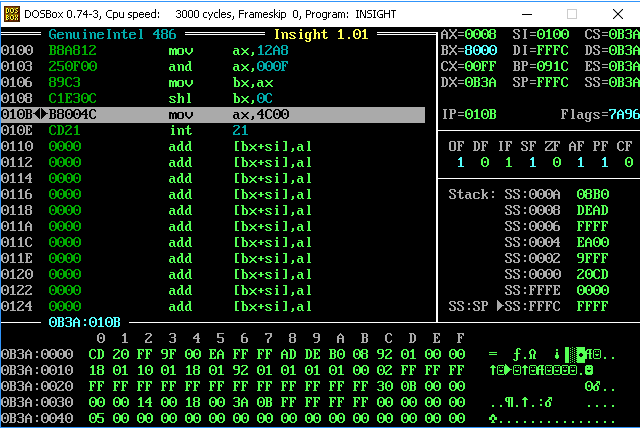
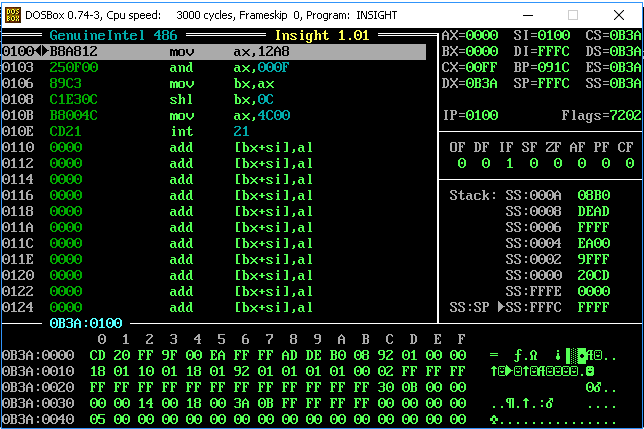
;---[ Стандартное завершение программы ]----------------------------------------

mov AX, 4C00h

int 21h

;===[ Начало сегмента данных ]==========================================

; ...



**Задача 5 (на один плюсик):**

Присвоить в регистры какие-либо небольшие начальные значения. Научиться вычислять выражения. Можно использовать регистры АХ, BX, CX, DX. Нельзя использовать стек и память. Нельзя заменять умножение многократным сложением/вычитанием, если разложение по степеням двойки короче. Здесь нужно придумать, как уложиться в четырех доступных регистрах.

DX = 4DX – 3BX + CX DX = 4DX – 3BX + CX – 5AX

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov AX, 1

mov BX, 1

mov CX, 1

mov DX, 1

shl DX, 2

shl BX, 1

add BX, CX

shl AX, 2

add AX, CX

sub dx, bx

add dx, cx

shl dx, 2

sub dx, bx

add dx, cx

sub dx, ax

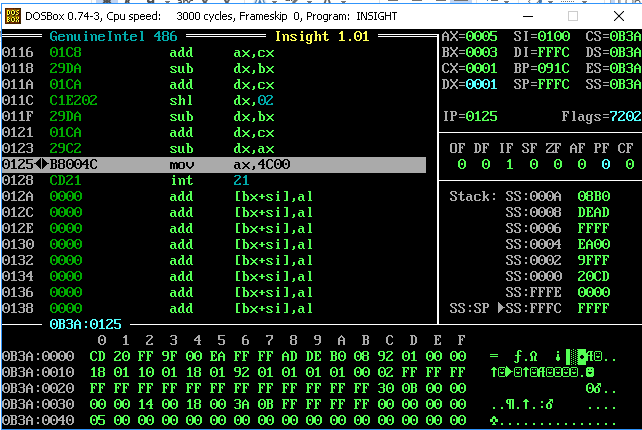
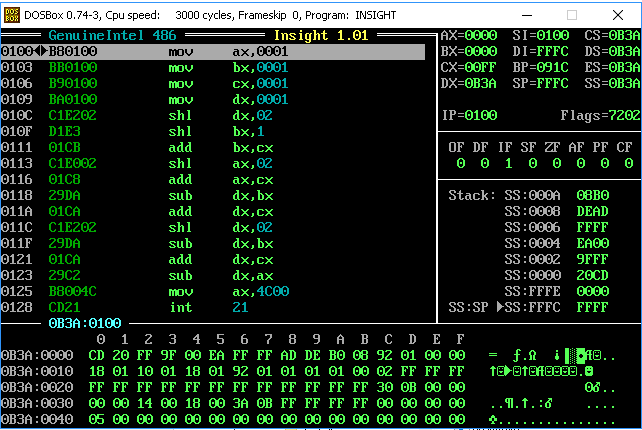
;---[ Стандартное завершение программы ]----------------------------------------

mov AX, 4C00h

int 21h

;===[ Начало сегмента данных ]==========================================

; ...



**Задача 6 (на один плюсик):**

Присвоить в регистр какое-либо начальное значение (например, 1234h). Научиться переставлять цифры местами, используя регистры, сдвиги, маски. Разумеется, все должно быть универсально, т.е. правильно работать для любого начального значения регистра.

заданы BX и СХ, поменять местами средние цифры ; BX=1234h, CX=ABCDh → BX=1BC4h, CX=A23Dh

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov bx, 1234h

mov cx, 0ABCDh

mov dx, bx

mov ax, cx

and bx, 0F00Fh

and cx, 0F00Fh

and ax, 0FF0h

and dx, 0FF0h

or cx, dx

or bx, ax

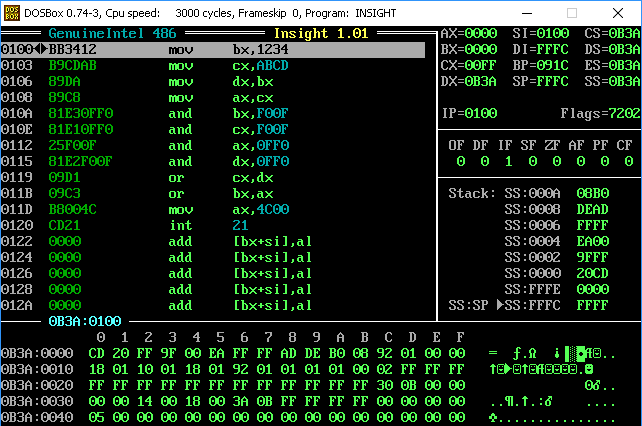
;---[ Стандартное завершение программы ]----------------------------------------

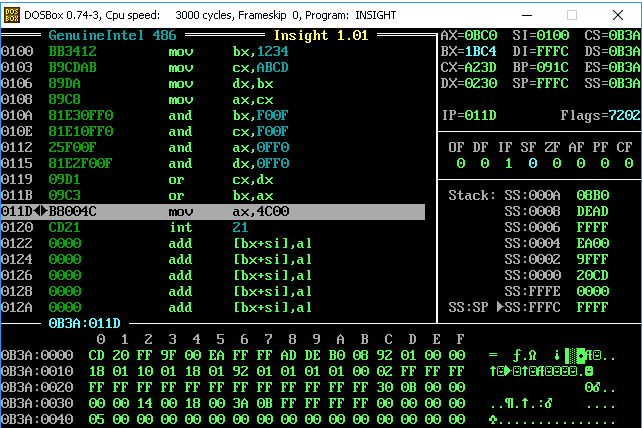
mov AX, 4C00h

int 21h

;===[ Начало сегмента данных ]==========================================

; ...





**Задача 7 (на один плюсик):**

Обнулить регистр. Далее создать оператор цикла с параметром, не более чем на 16 итераций. Внутри тела цикла разрешены только сдвиги **SHL** и инкременты **INC**. Добиться, чтобы единичные биты постепенно продвигались справа налево, пока не получится требуемая картинка. Нельзя делать в теле цикла несколько инкрементов подряд ради переноса единицы в старший разряд. Нельзя делать никаких доработок после цикла – он должен сам уметь вовремя остановиться.

(Пример: 1111111111111111b – это цикл на 16 итераций, в теле выполняется 1 сдвиг и 1 инкремент)

DX →0101 0101 0101 0101b = 5555h

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov dx, 0

mov cx, 8

M: shl dx, 2

inc dx

loop M

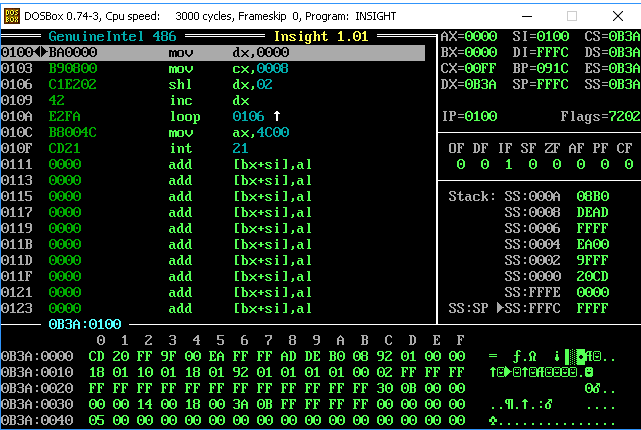
;---[ Стандартное завершение программы ]----------------------------------------

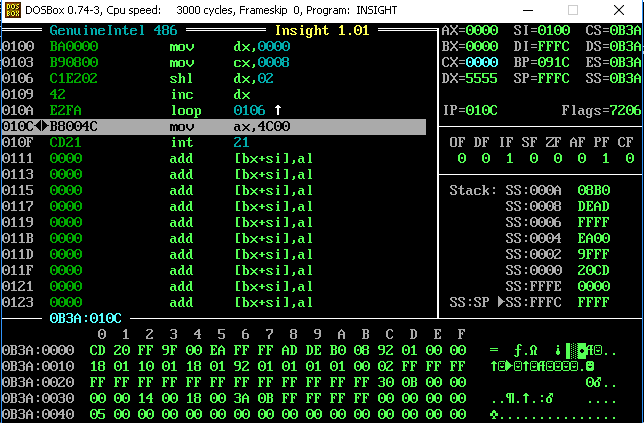
mov AX, 4C00h

int 21h

;===[ Начало сегмента данных ]==========================================

; ...





**Задача 8 (на один плюсик):**

Присвоить в регистры какие-либо небольшие начальные значения. Реализовать условный оператор

Заданы AX и BX, если (AX+BX)<1Fh, то CX = 1111h, иначе CX = FFFFh.

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov AX, 40

mov BX, 40

add AX, BX

CMP AX, 1Eh

JS NO

YES:

MOV CX, 0FFFFh

NO:

MOV CX, 1111h

JMP FIN

FIN:

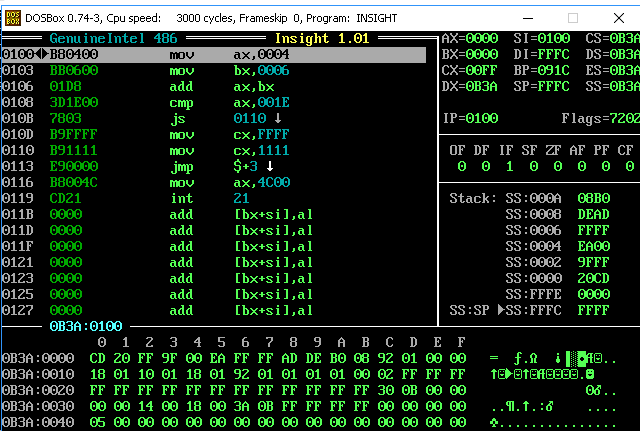
;---[ Стандартное завершение программы ]----------------------------------------

mov AX, 4C00h

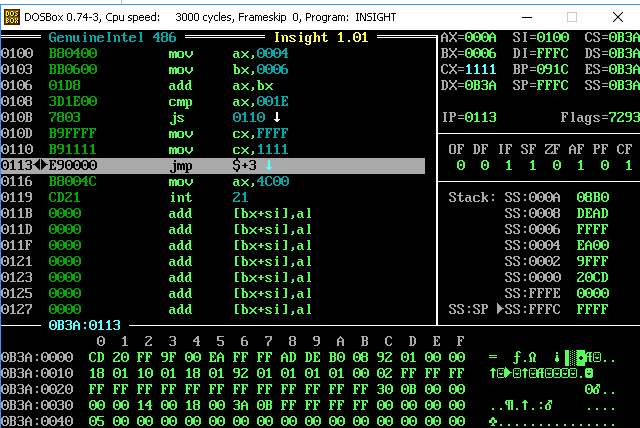
int 21h

;===[ Начало сегмента данных ]==========================================

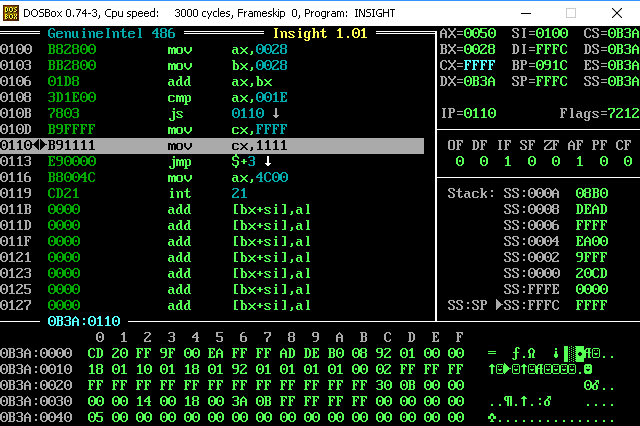
; ...



**ЕСЛИ УСЛОВИЕ ИСТИННО**



**ЕСЛИ УСЛОВИЕ ЛОЖНО**



**Задача 9 (на два плюсика):**

Присвоить в регистры какие-либо небольшие начальные значения. Реализовать составной условный оператор.

Заданы AX, BX и CX, если (AX≥3) and (BX=CX) то DX = 1111h, иначе DX = FFFFh

;===[ Начало сегмента кода ]============================================

MYCODE: segment .code

org 100h ; Обязательная директива ТОЛЬКО для COM-файлов

START: ;---[ Точка старта ]---------------------------------------------------------------------

mov AX, 4

mov BX, 1

mov CX, 2

CMP AX, 3

JS NO

CMP BX, CX

JS NO

JNS YES

YES:

MOV DX, 1111h

JMP FIN

NO:

MOV DX, 0FFFFh

FIN:

;---[ Стандартное завершение программы ]----------------------------------------

mov AX, 4C00h

int 21h

;===[ Начало сегмента данных ]==========================================

; ...

**ЕСЛИ ПЕРВОЕ УСЛОВИЕ ВЕРНО А ВТОРОЕ ЛОЖНО**

