«Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Боровцов Е.Г.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Отчёт

Лабораторной работе №2

«Cистема команд процессоров и методы адресации»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Боровцов Е.Г.

Барнаул 2020

Цель лабораторной работы:

Цель данной лабораторной работы состоит в дальнейшем изучении особенностей

выполнения команд процессоров различных групп и знакомстве с методами адресации

памяти при разработке программ на машинном языке.

Задание к лабораторной работе:- изучить соответствующий теоретический материал, используя конспекты и

литературу;

- на примере функциональной модели ЭВМ-2 (приложение 2) ознакомиться с

системой команд процессора и особенностями выполнения команд различных групп

(особое внимание уделить реализации команд умножения и деления, переходов и

организации цикла со счетчиком);

- рассмотреть допустимые в системе команд формы адресации и их реализацию;

- составить программы для реализации заданных выражений в кодах модели или

используя язык ассемблера модели, при этом учесть то, что программа для реализации

линейного выражения должна быть реализована тремя способами - с использованием

абсолютной формы адресации, с использованием относительной формы и с

использованием только регистров и стэковой памяти; программа для реализации

ветвящегося процесса должна обязательно использовать команды сравнения;

циклическая программа должна быть реализована любыми двумя из трех способов - по

схеме цикла с предусловием, цикла с постусловием, и цикла со счетчиком;

- ввести полученные программы в моделируемую память; выполнить программы

для различных вариантов исходных данных, наблюдая, как изменяется состояние

компонентов процессора ЭВМ и памяти при выполнении каждой команды; разобраться,

почему и как модифицируются те или иные объекты, как выполняются команды и как

осуществляется доступ к операндам;

- составить отчет по лабораторной работе;

Заданное выражение: 

1. Абсолютная адресация

in 0 ;ввод числа 'A' из порта с адресом 0 в РОН a

stor a,A ;запомнить число 'A' по символическому адресу A

in 1 ;ввод числа 'B' из порта с адресом 1 в РОН a

stor a,B ;запомнить число 'B' по символическому адресу B

in 2 ;ввод числа 'C' из порта с адресом 2 в РОН a

stor a,C ;запомнить число 'C' по символическому адресу C

in 3 ;ввод числа 'D' из порта с адресом 3 в РОН a

stor a,D ;запомнить число 'D' по символическому адресу D

load b,A ;в rB загрузить 'A'

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,B ;(rA,rB)/B-> частное в rB, остаток в rA

stor b,tmp ;запомнить результат деления по адресу tmp

load b,D ;в rB загрузить 'D'

mum a,D ;(rB)\*(F)-> (rA,rB)

stor b,tmp2 ;запомнить результат умножения по адресу tmp2

load b,C ;в rB загрузить 'C'

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,tmp2 ;(rA,rB)/B-> частное в rB, остаток в rA

stor b,tmp2 ;запомнить результат деления по адресу tmp2

load a,tmp ;в rA загрузить 'tmp'

SBR a,b ;вычесть из (rA) (rB)

out 7 ;вывод в 7 поле

stop ;стоп

A: .ds 1 ;ячейка (1 зарезервированный байт) для A

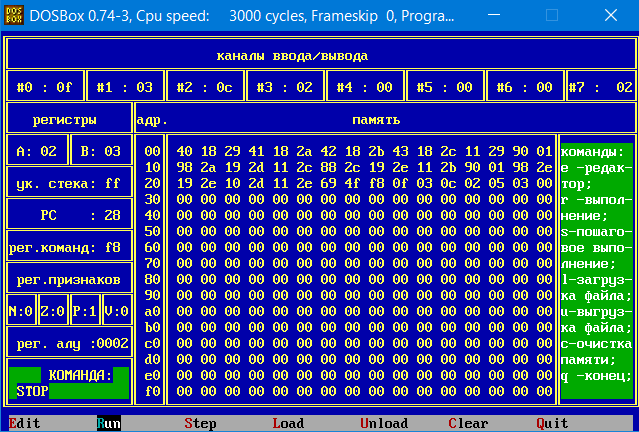
B: .ds 1 ;ячейка (1 зарезервированный байт) для B

C: .ds 1 ;ячейка (1 зарезервированный байт) для C

D: .ds 1 ;ячейка (1 зарезервированный байт) для D

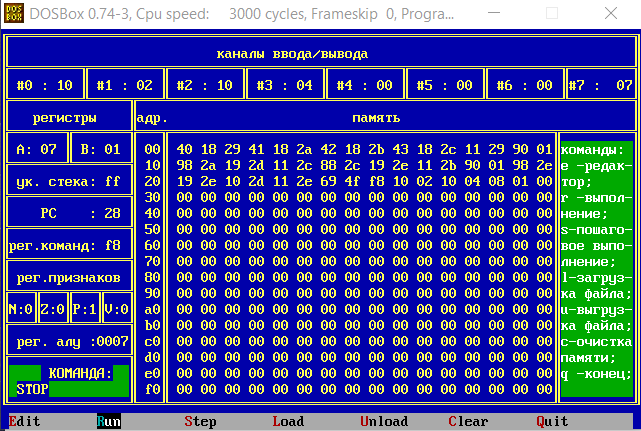
tmp: .ds 1 ;ячейка (1 зарезервированный байт) для tmp

tmp2: .ds 1 ;ячейка (1 зарезервированный байт) для tmp2



16-ая СС: f/3-c/(2\*2)=2

10-ая СС: 15/3-12/(2\*2)=5-3=2



16-ая СС: 10/2-10/(4\*4)=7

10-ая СС: 16/2-16/(4\*4)=8-1=7

1. Относительная адресация

.base b ; rB - базовый

start: mvi b,start; Относительный адрес - в rB: rB - базовый

in 0 ;ввод числа 'A' из порта с адресом 0 в РОН a

stor a,A ;запомнить число 'A' по символическому адресу A

in 1 ;ввод числа 'B' из порта с адресом 1 в РОН a

stor a,B ;запомнить число 'B' по символическому адресу B

in 2 ;ввод числа 'C' из порта с адресом 2 в РОН a

stor a,C ;запомнить число 'C' по символическому адресу C

in 3 ;ввод числа 'D' из порта с адресом 3 в РОН a

stor a,D ;запомнить число 'D' по символическому адресу D

push b ;(rB)v (sp) сохранить базовый адpес в стеке

load a,B ;в rA загрузить 'A'

push a ;(rB)v (sp) сохранить базовый адpес в стеке

load b,A ; загpузить в rB A, используя относительный адpес

mui a,1 ; pаспpостpанить знак

dvm a,1(sp) ; pазделить (rA,rB) на содержимое веpхушки стека

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rB) : вытолкнуть из стека в rB B

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

stor a,tmp ; запомнить по относительному адресу результат A/B

push b ;(rB)v (sp) сохранить базовый адpес в стеке

load b,D ;в rB загрузить 'D'

mur a,b ;(rB)\*(rB)-> (rA,rB)

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

push b ; вернуть 0 в стек

push a ; положить D\*D в стек

load b,c ; загрузить ‘C’ в rB

mui a, 1 ;расширить знак(преобразовать слово в двойное слово)

dvm a,1(sp) ;(rA,rB)/B-> частное в rB, остаток в rA

mov a,b ; пеpеслать частное в rA

pop b ; (sp)^(rB) : вытолкнуть из стека в rB B

pop b ; (sp)^(rb) : восстановить из стека базовый адрес

push b ; вернуть 0 в стек

push a ; положить C/(D\*D) в стек

load a,tmp ;в rA загрузить 'tmp'

pop b ; вытянуть C/(D\*D) в rB

sbr a,b ;вычесть из (rA) (rB)

out 7 ;вывод в 7 поле

stop ;стоп

A: .ds 1 ;ячейка (1 зарезервированный байт) для A

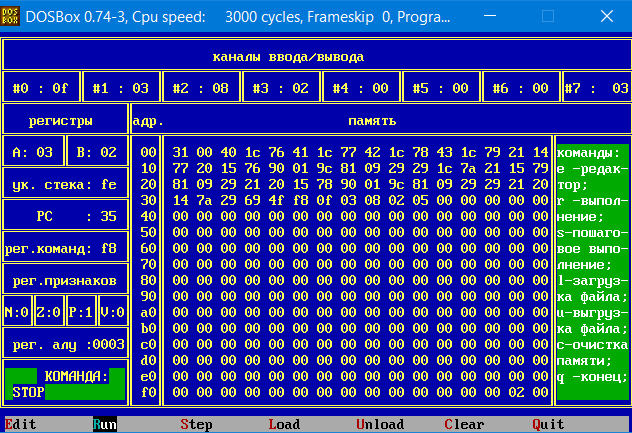
B: .ds 1 ;ячейка (1 зарезервированный байт) для B

C: .ds 1 ;ячейка (1 зарезервированный байт) для C

D: .ds 1 ;ячейка (1 зарезервированный байт) для D

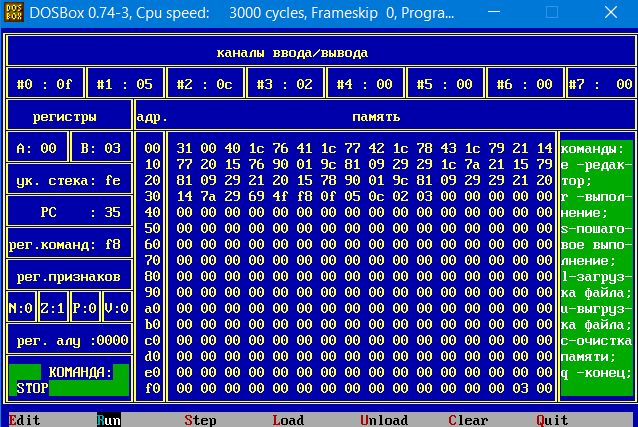
tmp: .ds 1 ;ячейка (1 зарезервированный байт) для tmp

tmp2: .ds 1 ;ячейка (1 зарезервированный байт) для tmp2



16-ая СС: f/3-8/(2\*2)=3

10-ая СС: 15/3-8/(2\*2)=5-2=3



16-ая СС: f/5-c/(2\*2)=0

10-ая СС: 15/5-12/(2\*2)=3-3=0

1. С использованием регистра и стековой памяти

in 0 ;ввод D

mov b,a ;скопировать rA в rB

mur a,a ;rB\*rA D\*D

push b ;D\*D в стек

in 1 ;ввод C

mov b,a ;скопировть rA в rB

mui a,1 ;распространение знака

dvm a,1(sp) ;dB(C)/вершину стека(D\*D)

push b ;C/(D\*D) в стек

in 2 ;ввод B

push a ;B в стек

in 3 ;ввод A

mov b,a ;скопировть rA в rB

mui a,1 ;распространение знака

dvm a,1(sp) ;dB(A)/вершину стека(B)

pop a ;вытолкнуть B из стека

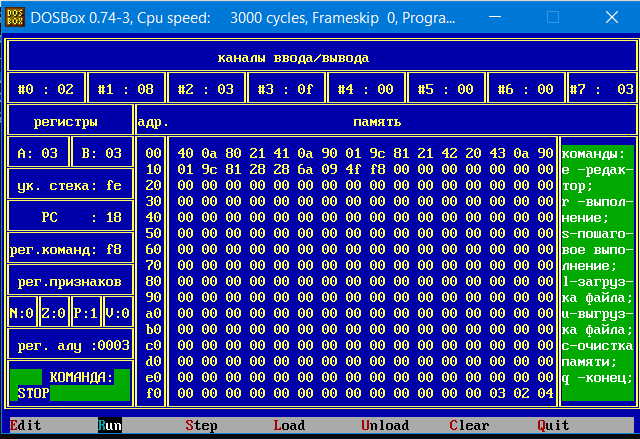
pop a ;вытолкнуть C/(D\*D) из стека

sbr b,a ;rB-rA

mov a,b ;скопировть rB в rA

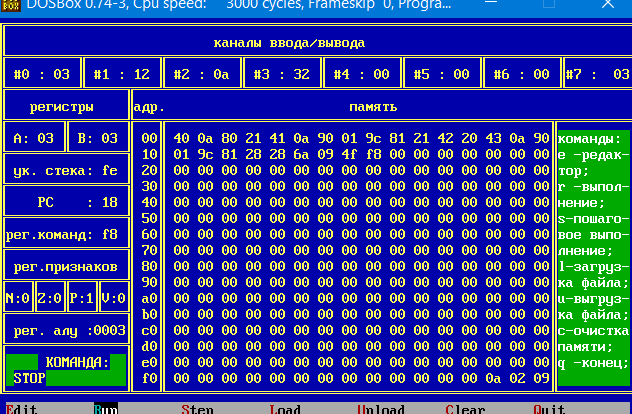
out 7 ;вывести в 7 порт

stop ;останов



16-ая СС: f/3-8/(2\*2)=3

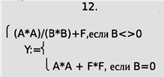
10-ая СС: 15/3-8/(2\*2)=5-2=3

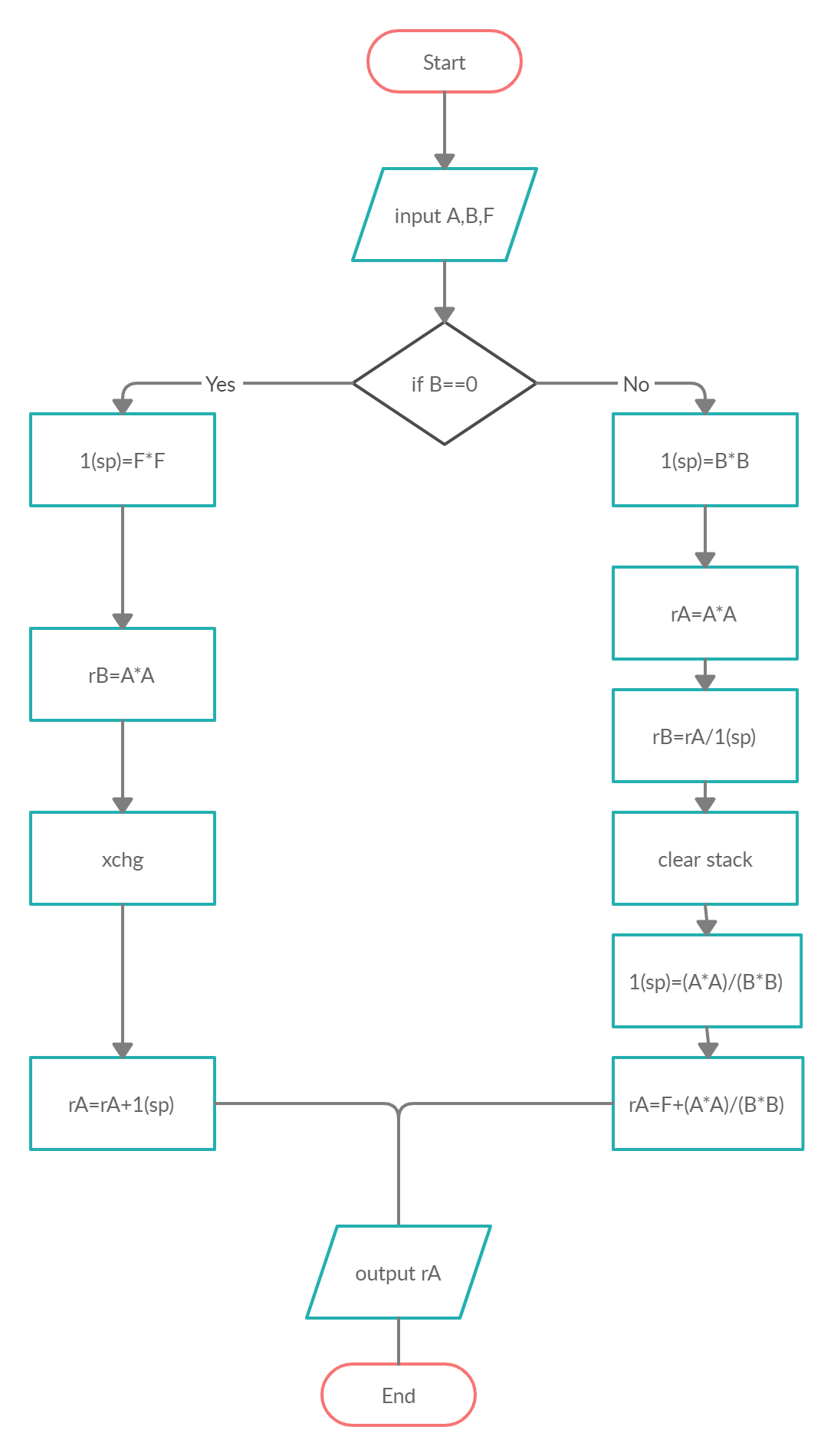


16-ая СС: 32/a-12/(3\*3)=3

10-ая СС: 50/10-18/(3\*3)=5-2=3

1. **Применение команд сравнений и переходов**





in 0 ;ввод A

stor a,A ;запомнить в A

in 1 ;ввод F

stor a,F ;запомнить в F

in 2 ;ввод B

stor a,B ;запомнить в B

cmi a,0 ;сравнить (rA) c 0 (установить регистр флагов)

jz label1 ;переход, если установлен признак Z (X==0)

xchg ;обмен содержимого rA и rB

mur a,b ;rB\*rB результат в rB (B\*B)

push b ;B\*B в стек

xchg ;обмен содержимого rA и rB

load b,A ;загрузить A в rB

mur a,b ;rB\*rB результат в rB (A\*A)

mui a,1 ;распространение знака

dvm a,1(sp) ;(rA,rB)/((sp)+1),частное в rB, остаток в rA : C

pop a ;очистить стек

push b ;положить (A\*A)/(B\*B) в стек

load a,F ;загрузить F в rA

adm a,1(sp) ;сложить rA и загрузить A в rB

jmp label2 ;безусловный переход на вывод

label1: load b,F;загрузить F в rB

mur a,b ;rB\*rB результат в rB (F\*F)

push b ;F\*F в стек

load b,A ;загрузить A в rB

mur a,b ;rB\*rB результат в rB (A\*A)

xchg ;

adm a,1(sp) ;

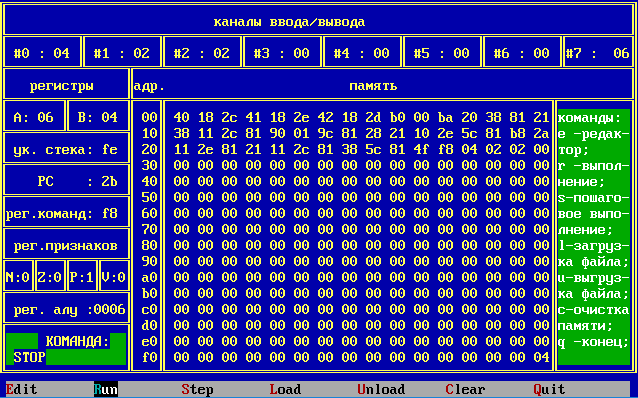
label2: out 7 ;вывод в 7 порт

stop ;

A: .ds 1 ; ячейки

B: .ds 1 ; для

F: .ds 1 ; хранения

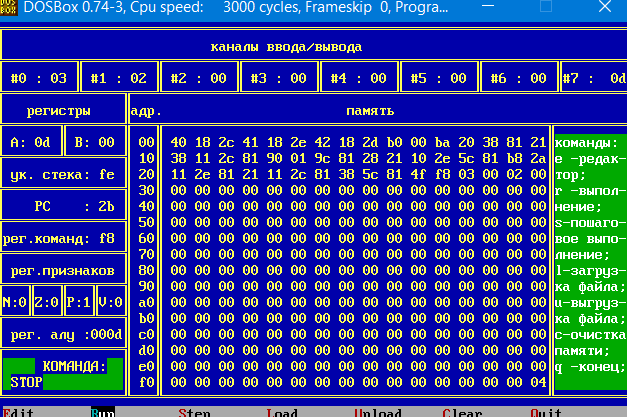
****

16-ая СС: A=4 B=2 F=2

* (4\*4)/(2\*2)+2=6

10-ая СС: A=4 B=2 F=2

* (4\*4)/(2\*2)+2=16/4+2=6



16-ая СС: A=3 B=0 F=2

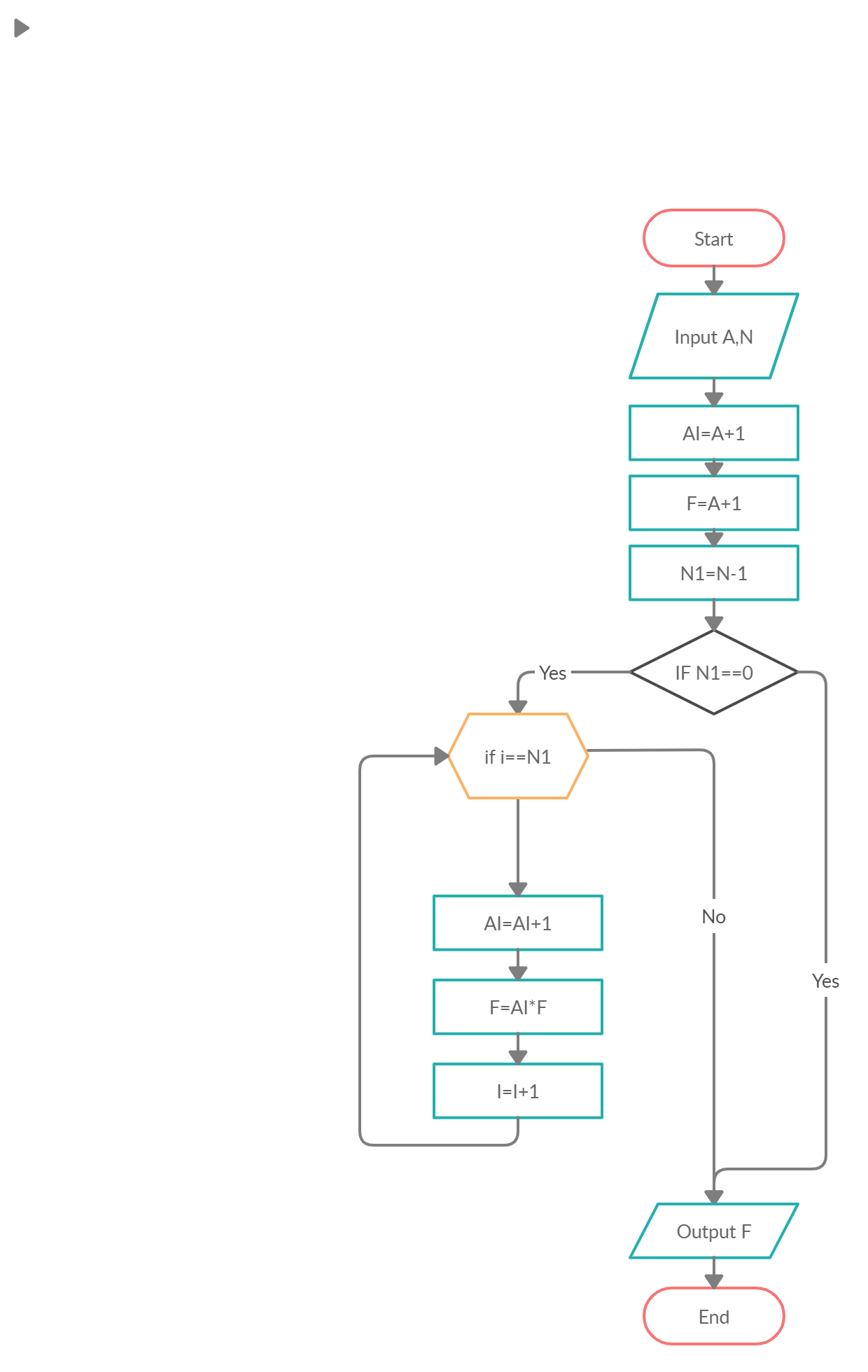
* (3\*3)+(2\*2)=d

10-ая СС: A=3 B=0 F=2

* (3\*3)+(2\*2)=13



1. Условный, безусловный



mvi b,1 ; непосредственная загрузка 1 в rB

stor b,I ; i:=1

in 0 ; ввод A

stor a,A ; a:=A

adr a,b ; A+1

stor a,AI ; AI:=A+1 (AI -текущее значение A+i)

stor a,F ; F:=A+1 текущее значение функции (результат при N=2)

in 1 ; ввод N в rA

sbi a,1 ; N-1

stor a,N1 ; n1 = n-1

cmi a,1 ; сравнить (rA) с 0 (n-1=1 ?)

jz exit ; перейти по символическому адресу exit при (rA)=0

xchg ; (rA)-(rB) : в rA - 1(i:=1); в rB - N-1

cycl: load b,N1 ; загрузить в rB N

load a,I ; загрузить в rA I

cmr a,b ; в соответствии с операцией (R1)-(R2)

jz exit ; переход по признаку P (I==n-1)

load a,AI ; в rA текущее значение функции

mvi b,1 ; непосредственная загрузка 1 в rB

adr a,b ; (A+i)+1

stor a,AI ; AI:=(A+i)+1(AI -текущее значение A+i)

load b,F ; в rB текущее значение функции

mur a,a ; (rB)\*(rA)->(rA,rB) (A+i)\*F

stor b,F ; F=(A+i)\*F

load b,I ; I в rB

adi b,1 ; I+1

stor b,I ; I=I+1

jmp cycl ; переход на проверку условия окончания цикла

exit: load a,F ; (rA)<- результат вычисления выражения

out 7 ; вывод результата в порт 1

stop ; стоп

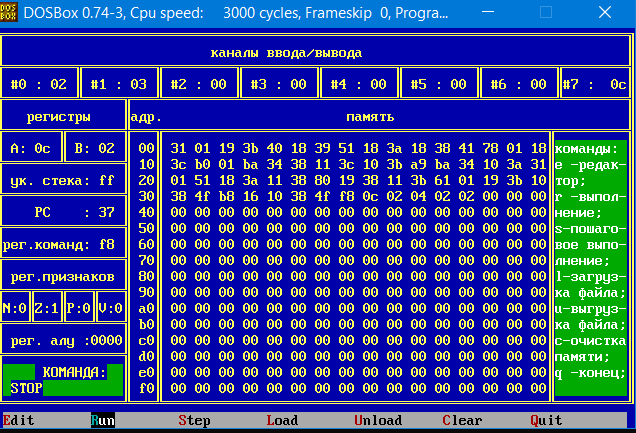
F: .ds 1 ; ячейка под текущее значение функции

A: .ds 1 ; ячейка под A

AI: .ds 1 ; ячейка под текущее значение A+i

I: .ds 1 ; ячейка под текущее i

N1: .ds 1 ; ячейка для хранения N-1

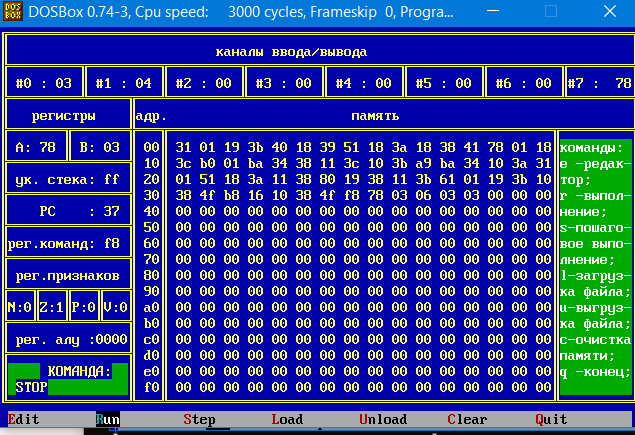


16-ая СС: A=2 N=3

* (2+1)\*(2+2)=c

10-ая СС: A=2 N=3

* (2+1)\*(2+2)=12



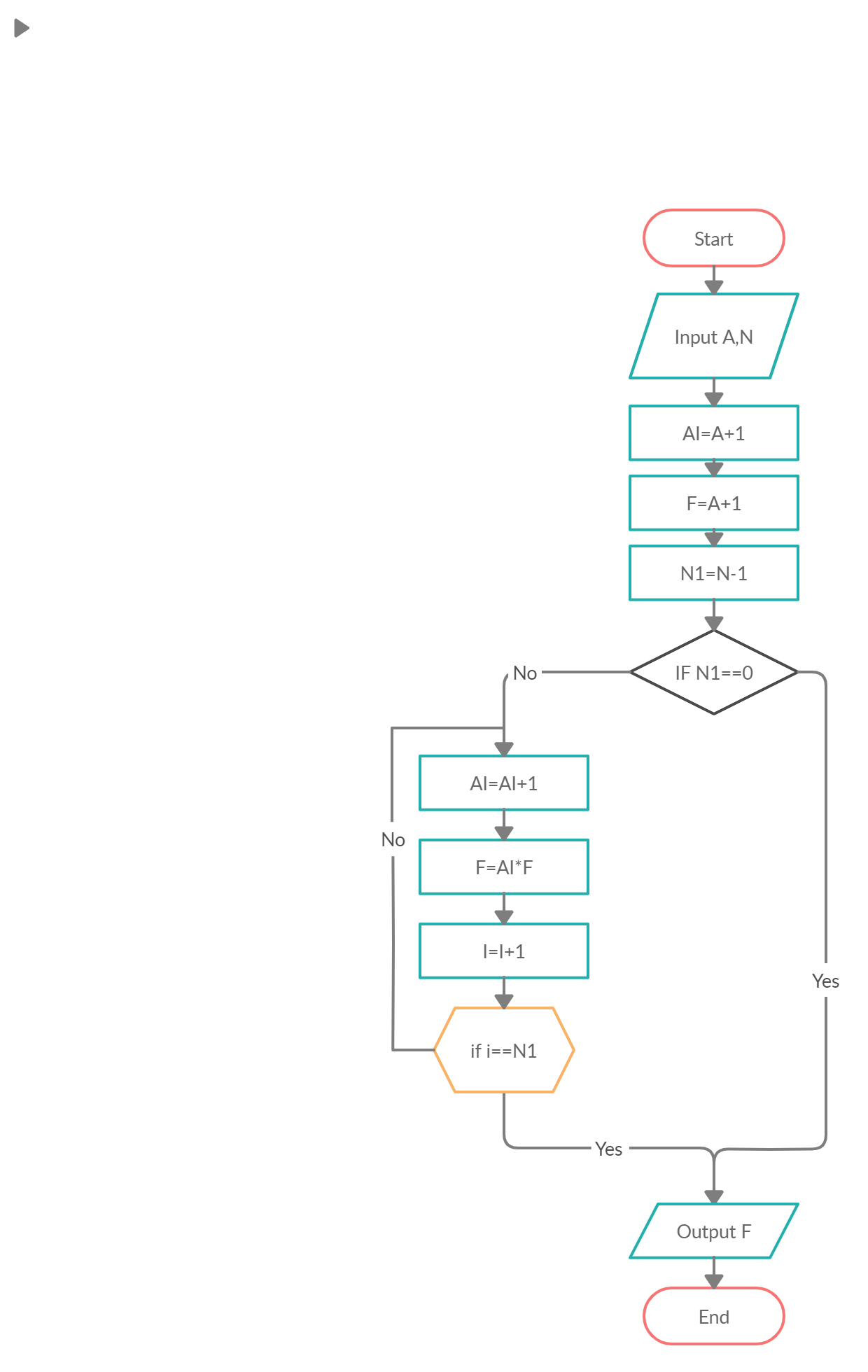
16-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=78

10-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=4\*5\*6=120

1. Постоянное условие



mvi b,1 ; непосредственная загрузка 1 в rB

stor b,I ; i:=1

in 0 ; ввод A

stor a,A ; a:=A

adr a,b ; A+1

stor a,AI ; AI:=A+1 (AI -текущее значение A+i)

stor a,F ; F:=A+1 текущее значение функции (результат при N=2)

in 1 ; ввод N в rA

sbi a,1 ; N-1

stor a,N1 ; n1 = n-1

cmi a,1 ; сравнить (rA) с 0 (n-1=1 ?)

jz exit ; перейти по символическому адресу exit при (rA)=0

xchg ; (rA)-(rB) : в rA - 1(i:=1); в rB - N-1

cycl: load a,AI ; в rA текущее значение функции

mvi b,1 ; непосредственная загрузка 1 в rB

adr a,b ; (A+i)+1

stor a,AI ; AI:=(A+i)+1(AI -текущее значение A+i)

load b,F ; в rB текущее значение функции

mur a,a ; (rB)\*(rA)->(rA,rB) (A+i)\*F

stor b,F ; F=(A+i)\*F

load b,I ; I в rB

adi b,1 ; I+1

stor b,I ; I=I+1

load b,N1 ; загрузить в rB N

load a,I ; загрузить в rA I

cmr a,b ; в соответствии с операцией (R1)-(R2)

jn cycl ; переход по признаку P (I==n-1)

exit: load a,F ; (rA)<- результат вычисления выражения

out 7 ; вывод результата в порт 1

stop ; стоп

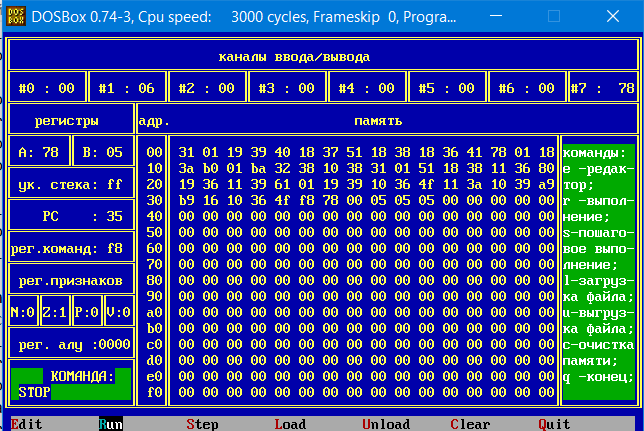
F: .ds 1 ; ячейка под текущее значение функции

A: .ds 1 ; ячейка под A

AI: .ds 1 ; ячейка под текущее значение A+i

I: .ds 1 ; ячейка под текущее i

N1: .ds 1 ; ячейка для хранения N-1

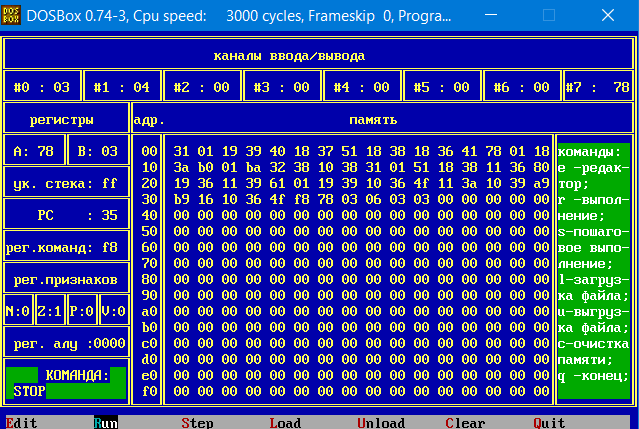


16-ая СС: A=0 N=6

* 1\*2\*3\*4\*5=78

10-ая СС: A=0 N=6

* 1\*2\*3\*4\*5=120



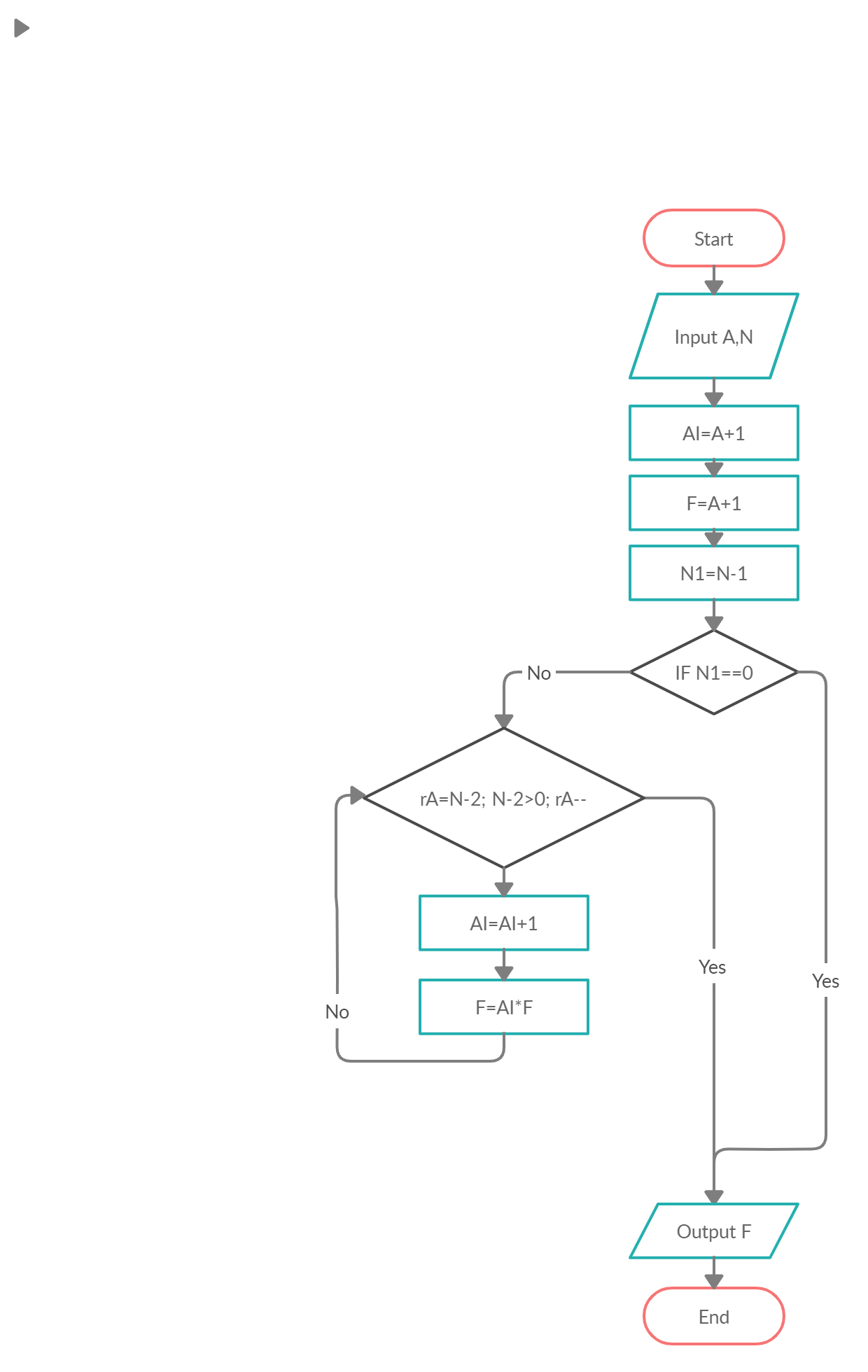
16-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=78

10-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=4\*5\*6=120

1. Команда организации цикла со счетчиком



mvi b,1 ; непосредственная загрузка 1 в rB

stor b,I ; i:=1

in 0 ; ввод A

stor a,A ; a:=A

adr a,b ; A+1

stor a,AI ; AI:=A+1 (AI -текущее значение A+i)

stor a,F ; F:=A+1 текущее значение функции (результат при N=2)

in 1 ; ввод N в rA

sbi a,1 ; N-1

stor a,N1 ; n1 = n-1

cmi a,1 ; сравнить (rA) с 0 (n-1=1 ?)

jz exit ; перейти по символическому адресу exit при (rA)=0

sbi a,1 ; rA=N-2

cycl: push a ; сохранить rA в стеке(счетчик цикла)

load a,AI ; в rA текущее значение функции

mvi b,1 ; непосредственная загрузка 1 в rB

adr a,b ; (A+i)+1

stor a,AI ; AI:=(A+i)+1(AI -текущее значение A+i)

load b,F ; в rB текущее значение функции

mur a,a ; (rB)\*(rA)->(rA,rB) (A+i)\*F

stor b,F ; F=(A+i)\*F

pop a ;

loop a,cycl ;

exit: load a,F ; (rA)<- результат вычисления выражения

out 7 ; вывод результата в порт 1

stop ; стоп

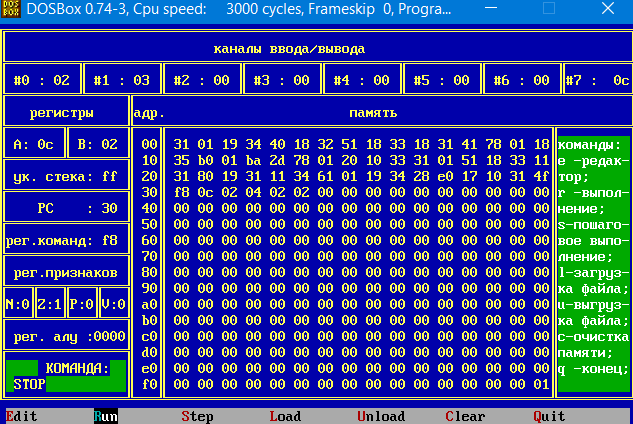
F: .ds 1 ; ячейка под текущее значение функции

A: .ds 1 ; ячейка под A

AI: .ds 1 ; ячейка под текущее значение A+i

I: .ds 1 ; ячейка под текущее i

N1: .ds 1 ; ячейка для хранения N-1

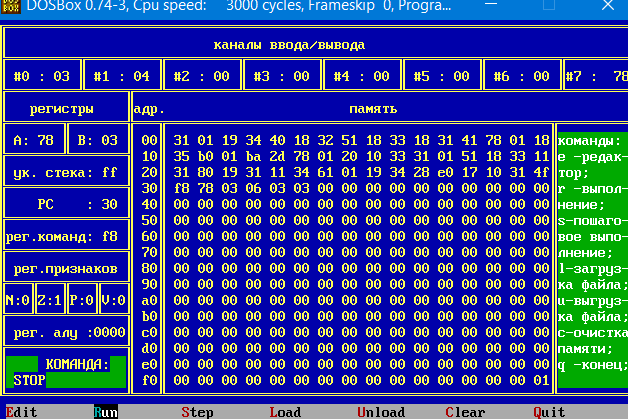


16-ая СС: A=2 N=3

* (2+1)\*(2+2)=c

10-ая СС: A=2 N=3

* (2+1)\*(2+2)=12



16-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=78

10-ая СС: A=3 N=4

* (3+1)\*(3+2)\*(3+3)=4\*5\*6=120