Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет (институт) Информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

А.И.Потупчик

(подпись преподавателя) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Отчет

по лабораторной (практической) работе №\_12\_

Знакомство с архитектурой ARM

(название лабораторной (практической) работы)

по дисциплине Архитектура ЭВМ

(наименование дисциплины)

ЛР 09.03.04.21.000 ОТ

(обозначение документа)

Студент группы ПИ-02 Р.А. Чередов

(инициалы, фамилия)

Преподаватель доцент, доцент А.И.Потупчик

(должность, ученое звание) (инициалы, фамилия)

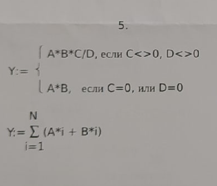
Барнаул 2022

**Задание:**

Используя систему команд архитектуры ARM CortexM3 и эмулятор соответствующей архитектуры, реализовать линейный, ветвящийся и циклический алгоритм.

При реализации учитывать особенности системы команд. Операции умножения и деления реализовать подпрограммами.

Y=a\*b+(c^2-d)



# Линейный алгоритм

Y=a\*b+(c^2-d)

**Тесты программы:**

number\_A DCD 2

number\_B DCD 2

number\_C DCD 2

number\_D DCD 2

result DCD 2

LDR R0,=number\_A

LDR R1,[R0,#0]

LDR R0,=number\_B

LDR R2,[R0,#0]

BL umn

MOV R5,R3

LDR R0,=number\_C

LDR R1,[R0,#0]

MOV R2,R1

BL umn

LDR R0,=number\_B

LDR R1,[R0,#0]

SUB R3,R3,R1

ADD R3,R3,R5

LDR R0,=result

STR R3,[R0,#0]

end

;

;

;

umn

LDR R3,=0 ; 0 -> R3

LDR R0,=0 ; 0 -> R0

LDR R4,=1 ; 1 -> R4

CMP R1,R0 ; Сравнить R1 и R0

CMPMI R2,R0 ; Сравнить r2 и 0 если r1 отрицательный

BMI A\_and\_B\_negativ ; Перейти по метке если r2 отрицательный

loop CMP R2,R0 ; Сравнить R2 и R0

ADDPL R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

SUBPL R2,R2,R4 ; R2 - R4 = R2 Если r2 положительный

SUBMI R3,R3,R1 ; R3 - R1 = R3 Если r2 отрицательный

ADDMI R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE loop ; Замкнуть цикл если R2 не равно 0

out MOV PC,LR ; Выход из подпрограммы

A\_and\_B\_negativ

ADD R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

ADD R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE A\_and\_B\_negativ ; Замкнуть цикл если R2 не равно 0

B out

**Входные данные:**

**A = 2, B = 2, C = 2, D = 2**

**Выходные данные:**

**Y=2\*2+(4-2)=6**



**Входные данные:**

**A = -2, B = 2, C = 2, D = 2**

**Выходные данные:**

**Y=-2\*2+(4-2)=-2**



# Ветвящийся алгоритм

Текст программы:

/\*

#### /A\*B\*C/D, если C<>0 и D<>0

**Y : = {**

#### \ A\*B, если C=0 или D=0

**number\_A DCD 2**

**number\_B DCD 2**

**number\_C DCD 2**

**number\_D DCD 2**

**result DCD 2**

**LDR R0, =number\_A**

**LDR R1,[R0,#0]**

**LDR R0, =number\_B**

**LDR R2,[R0,#0]**

**BL umn**

**MOV R5,R3**

**LDR R0, =number\_C**

**LDR R1,[R0,#0]**

**ldr r8,=0**

**CMP R1,R8**

**Beq exit**

**mov r2,r5**

**bl umn**

**mov r5,r3**

**mov r6,r3**

**ldr r0,=number\_D**

**ldr r2,[R0,#0]**

**CMP R2,R8**

**Beq exit**

**mov r1,r5**

**bl div**

**mov r5,r3**

**b exit1**

**exit ldr r0,=result**

**str R5,[R0,#0]**

**end**

**exit1**

**ldr r0,=result**

**str R6,[R0,#0]**

**end**

**umn**

LDR R3,=0 ; 0 -> R3

LDR R0,=0 ; 0 -> R0

LDR R4,=1 ; 1 -> R4

CMP R1,R0 ; Сравнить R1 и R0

CMPMI R2,R0 ; Сравнить r2 и 0 если r1 отрицательный

BMI A\_and\_B\_negativ ; Перейти по метке если r2 отрицательный

loop CMP R2,R0 ; Сравнить R2 и R0

ADDPL R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

SUBPL R2,R2,R4 ; R2 - R4 = R2 Если r2 положительный

SUBMI R3,R3,R1 ; R3 - R1 = R3 Если r2 отрицательный

ADDMI R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE loop ; Замкнуть цикл если R2 не равно 0

out MOV PC,LR ; Выход из подпрограммы

A\_and\_B\_negativ

ADD R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

ADD R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE A\_and\_B\_negativ ; Замкнуть цикл если R2 не равно 0

B out

div

LDR R3,=0 ; 0 -> R3

LDR R0,=0 ; 0 -> R0

LDR R4,=1 ; 1 -> R4

CMP R1,R0 ; Сравнить R1 и 0

CMPMI R2,R0 ; Если R1 отрицательный то сравнить R2 и 0

BMI negativ\_A\_and\_B ; Если R2 отрицательный то переход по метке

CMP R1,R0 ; Сравнить R1 и R0

BMI negativ\_A ; Если R1 отрицательный то переход по метке

CMP R2,R0 ; Сравнить R2 и R0

BMI negativ\_B ; Если R1 отрицательный то переход по метке

positive

SUB R1,R1,R2 ; R1 - R2 = R1

ADD R3,R3,R4 ; R3 + 1 = R3

CMP R1,R2 ; Сравнить R1 и R2

BPL positive ; Замкнуть цикл если R1 >= 0

CMP R1,R0 ; Сравнить R1 и 0

SUBNE R3,R3,R4 ; Если R1 не равен 0 то R3 - 1

B exit ; Перейти к выходу из подпрограммы

negativ\_A

ADD R1,R1,R2 ; R1 + R2 = R1

SUB R3,R3,R4 ; R3 - 1 = R3

CMP R1,R0 ; Сравнить R1 и R0

BMI negativ\_A ; Замкнуть цикл если R1 < 0

CMP R1,R0 ; Сравнить R1 и 0

ADDNE R3,R3,R4 ; Если R1 не равен 0 то R3 + 1

B exit ; Перейти к выходу из подпрограммы

negativ\_B

ADD R1,R1,R2 ; R1 + R2 = R1

SUB R3,R3,R4 ; R3 - 1 = R3

CMP R1,R0 ; Сравнить R1 и R0

BPL negativ\_B ; Замкнуть цикл если R1 >= 0

CMP R1,R0 ; Сравнить R1 и R0

ADDNE R3,R3,R4 ; Если R1 не равен 0 то R3 + 1

B exit ; Перейти к выходу из подпрограммы

negativ\_A\_and\_B

SUB R1,R1,R2 ; R1 - R2 = R1

ADD R3,R3,R4 ; R3 + 1 = R3

CMP R1,R0 ; Сравнить R1 и R0

BMI negativ\_A\_and\_B ; Замкнуть цикл если R1 < 0

CMP R1,R0 ; Сравнить R1 и R0

SUBNE R3,R3,R4 ; Если R1 не равен 0 то R3 - 1

B out ; Перейти к выходу из подпрограммы

**Тесты программы:**

**Входные данные:**

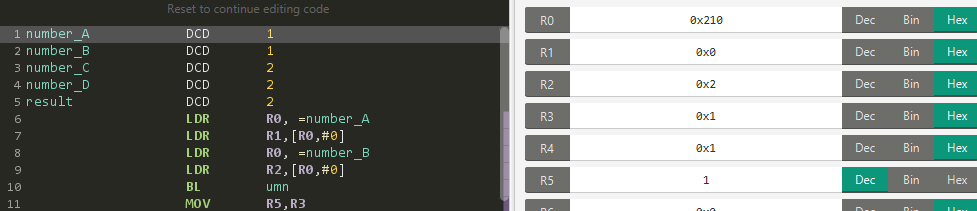
**A = 1, B = 1, C = 2, D = 2**

**Выходные данные:**

**С < > 0 и D < > 0**

**Y = A\*B\*C / D= 1\*1\*2/ 2 = 2 / 2 = 1**

Результат работы:



**Входные данные:**

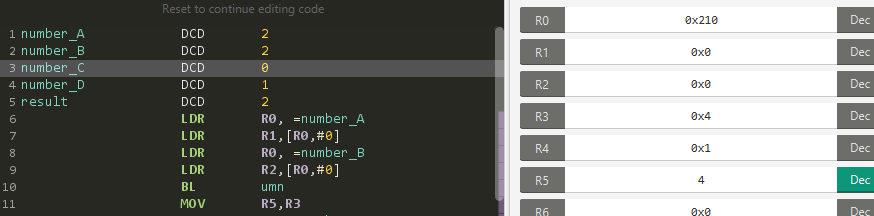
**A = 2, B = 2, C = 0, D = 1**

**Выходные данные:**

**C=0 или D=0**

**Y = A\*B= 2\*2=4**

Результат работы:



**Входные данные:**

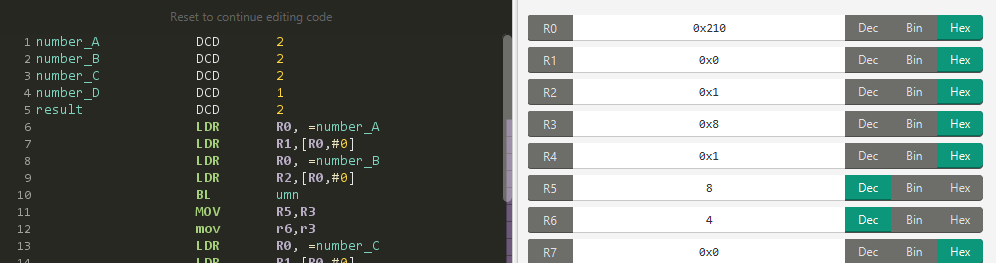
**A = 2, B = 2, C = 2, D = 0**

**Выходные данные:**

**C=0 или D=0**

**Y = A\*B= 2\*2=4**

Результат работы: в данном случае перейдет на exit1 и результат будет в ячейке r6



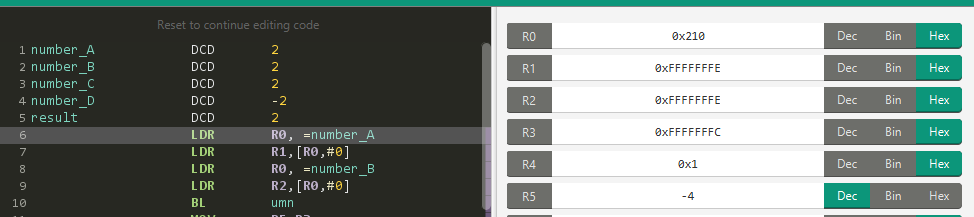
**Входные данные:**

**A = 2, B = 2, C = 2, D = -2**

**Выходные данные:**

**Y = A\*B= 2\*2\*2/-2=-4**

Результат работы:



## Циклический алгоритм

Текст программы:

number\_A DCD 2

number\_B DCD 2

number\_N DCD 1

result DCD 2

LDR R0, =number\_A

LDR R9,[R0,#0]

LDR R0, =number\_B

LDR R7,[R0,#0]

LDR R0, =number\_N

LDR R6,[R0,#0]

ldr r8,=1

maincycl mov r2,r6

mov r1,r9

bl umn

add r5,r3,r5

mov r1,r7

mov r2,r6

bl umn

add r5,r3,r5

sub r6,r6,r8

ldr r0,=0

cmp r6,r0

bne maincycl

ldr r0,=result

str r5,[R0,#0]

end

umn

LDR R3,=0 ; 0 -> R3

LDR R0,=0 ; 0 -> R0

LDR R4,=1 ; 1 -> R4

CMP R1,R0 ; Сравнить R1 и R0

CMPMI R2,R0 ; Сравнить r2 и 0 если r1 отрицательный

BMI A\_and\_B\_negativ ; Перейти по метке если r2 отрицательный

loop CMP R2,R0 ; Сравнить R2 и R0

ADDPL R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

SUBPL R2,R2,R4 ; R2 - R4 = R2 Если r2 положительный

SUBMI R3,R3,R1 ; R3 - R1 = R3 Если r2 отрицательный

ADDMI R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE loop ; Замкнуть цикл если R2 не равно 0

out MOV PC,LR ; Выход из подпрограммы

A\_and\_B\_negativ

ADD R3,R3,R1 ; R3 + R1 = R3 Если r2 положительный

ADD R2,R2,R4 ; R2 + R4 = R2 Если r2 отрицательный

CMP R2,R0 ; Сравнить R2 и R0

BNE A\_and\_B\_negativ ; Замкнуть цикл если R2 не равно 0

B out

**Тесты программы:**

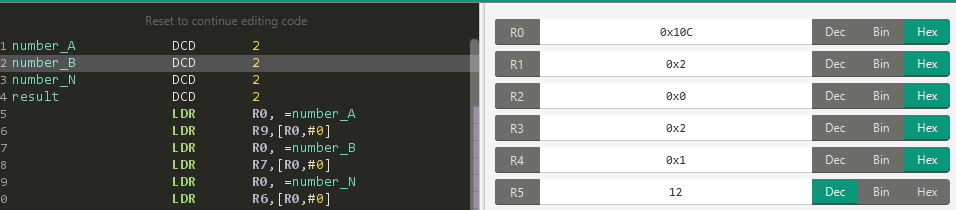
**Входные данные:**

**A = 2, B = 2, N = 2**

### Выходные данные:

**Y=2\*2+2\*2+2\*1+2\*1=12**

Результат работы:



**Тесты программы:**

**Входные данные:**

**A = 2, B = 2, N = 3**

### Выходные данные:

**Y=2\*3+2\*3+2\*2+2\*2+2\*1+2\*1=24**

Результат работы: 