**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Организация ЭВМ и систем»**

# Тема: Введение в ассемблер.

|  |  |
| --- | --- |
| Студент гр. 3381 | Лутфулин Д.А. |
| Преподаватель | Молодцев Д.А. |

Санкт-Петербург

2024

#### ****Цель**** работы

Создать программу на ассемблере, которая выполняет вычисление математического выражения, используя значения, размещённые в регистрах a2, a3 и a4. Результат вычислений должен быть сохранён в регистре a0.

#### ****Задание****

Ваше задание:

Напишите программу на ассемблере, которая вычисляет результат математического выражения в соответствии с вариантом. Убедитесь в корректности работы программы через автоматизированную систему.

Начальные данные на момент старта программы будут расположены в регистрах a2, a3, a4 соответственно. Результат выражения должен быть сохранен в регистр a0.

Весь код программы должен располагаться в метке solution. Программа должна заканчивать работу вызовом ret.

Шаблон программы для подготовки решения:

.globl solution

solution:

# a0 = result

ret

Ваше условие будет выведено ниже:

(a3 + ((-a3) \* (a4 | a2))) \* (a2 - (a4 \* (a4 + a4)))

Ваш seed = 5329281024

#### ****Выполнение**** работы

Программа разделена на две части: основная программа и подпрограмма solution. Основная программа инициализирует начальные значения регистров a2, a3, a4 и вызывает процедуру solution, где выполняется вычисление.

В ней выполняются следующие операции:

* Вычисление побитового ИЛИ между a4 и a2
* умножение -a3 на результат побитового ИЛИ
* удвоение a4 и умножение a4 на удвоенное значение
* вычитание произведения a4 на удвоенное значение из a2
* сложение a3 и результата умножения
* умножение результата сложения на разность, полученную на предыдущем шаге.

Результат сохраняется в регистре a0, который является итоговым значением выражения. В конце программы вызывается команда для завершения работы.

Результаты выполнения программы в пошаговом режиме:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Адрес инстр.** | **(Псевдо-) инстр.** | **Инструкция** | **16-ричный код инстр.** | **Содержимое регистров и ячеек памяти** | |
| **до вып. функции** | **после вып. функции** |
| 0 | li a2, 2 | addi x12 x0 2 | 00200613 | x12 0 0x00000000 | x12 2 0x00000002 |
| 4 | li a3, 3 | addi x13 x0 3 | 00300693 | x13 0 0x00000000 | x13 3 0x00000003 |
| 8 | li a4, 4 | addi x14 x0 4 | 00400713 | x14 0 0x00000000 | x14 4 0x00000004 |
| c | call solution | auipc x1 0x0 <\_start> | 00000097 | x1 0 0x00000000 | x1 12 0x0000000c |
| 10 | jalr x1 x1 16 | 010080e7 | x1 12 0x0000000c | x1 20 0x00000014 |
| 1c | or t0, a4, a2 | or x5 x14 x12 | 00c762b3 | x12 2 0x00000002  x14 4 0x00000004  x5 0x00000000 | x12 2 0x00000002  x14 4 0x00000004  x5 0x00000006 |
| 20 | neg t1, a3 | sub x6 x0 x13 | 40d00333 | x13 3 0x00000003  x6 0 0x00000000 | x13 3 0x0000003  x6 0xfffffffd |
| 24 | mul t2, t1, t0 | mul x7 x6 x5 | 025303b3 | x7 0x00000000  x6 0xfffffffd  x5 0x00000006 | x7 -18 0xffffffee  x6 0xfffffffd  x5 0x00000006 |
| 28 | add t3, a4, a4 | add x28 x14 x14 | 00e70e33 | x14 4 0x00000004  x28 0 0x00000000 | x14 4 0x00000004  x28 8 0x00000008 |
| 2c | mul t4, a4, t3 | mul x29 x14 x28 | 03c70eb3 | x14 4 0x00000004  x28 8 0x00000008  x29 0 0x00000000 | x14 4 0x00000004  x28 8 0x00000008  x29 32 0x00000020 |
| 30 | sub t5, a2, t4 | sub x30 x12 x29 | 41d60f33 | x12 2 0x00000002  x29 32 0x00000020  x30 0 0x00000000 | x12 2 0x00000002  x29 32 0x00000020  x30 -30 0xffffffe2 |
| 34 | add t6, a3, t2 | add x31 x13 x7 | 00768fb3 | x7 -18 0xffffffee  x13 3 0x00000003  x31 0 0x00000000 | x7 -18 0xffffffee  x13 3 0x00000003  x31 -15 0xfffffff1 |
| 38 | mul a0, t6, t5 | mul x10 x31 x30 | 03ef8533 | x10 0 0x00000000  x31 -15 0xfffffff1  x30 -30 0xffffffe2 | x10 450 0x000001c2  x31 -15 0xfffffff1  x30 -30 0xffffffe2 |
| 3c |  | jalr x0 x1 0 | 00008067 |  |  |
| 14 | 00a00893 | addi x17 x0 10 | 00200613 | x17 0 0x00000000 | x17 0x0000000a |
| 18 | 00000073 | ecall | 00300693 |  |  |

**Вывод.**

Разработанная программа корректно вычисляет заданное математическое выражение и сохраняет результат в регистре a0.

# Приложение А Исходный код программы

.globl \_start

\_start:

# Инициализация начальных значений

li a2, 2 # a2 = 2

li a3, 3 # a3 = 3

li a4, 4 # a4 = 4

# Вызов решения

call solution

# Завершаем программу

li a7, 10 # Код завершения программы

ecall

.globl solution

solution:

# Сначала вычислим побитовое ИЛИ a4 | a2

or t0, a4, a2 # t0 = a4 | a2

# Теперь умножим -a3 на (a4 | a2)

neg t1, a3 # t1 = -a3

mul t2, t1, t0 # t2 = -a3 \* (a4 | a2)

# Далее вычислим (a4 + a4)

add t3, a4, a4 # t3 = a4 + a4

# Умножим a4 на (a4 + a4)

mul t4, a4, t3 # t4 = a4 \* (a4 + a4)

# Теперь вычислим (a2 - (a4 \* (a4 + a4)))

sub t5, a2, t4 # t5 = a2 - (a4 \* (a4 + a4))

# Теперь сложим a3 и (-a3 \* (a4 | a2))

add t6, a3, t2 # t6 = a3 + (-a3 \* (a4 | a2))

# Умножим (a3 + (-a3 \* (a4 | a2))) на (a2 - (a4 \* (a4 + a4)))

mul a0, t6, t5 # a0 = (a3 + (-a3 \* (a4 | a2))) \* (a2 - (a4 \* (a4 + a4)))

# Завершаем выполнение функции

ret