ГУАП

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | А.Н. Долидзе |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ВЫПОЛНЕНИЯ ЦЕЛОЧИСЛЕННОЙ ОПЕРАЦИИ ДЛЯ АРХИТЕКТУРЫ НАБОРА КОМАНД ARM. |
| по курсу: ОРГАНИЗАЦИЯ ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4941 |  |  |  | Н.С. Горбунов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы:** изучить набор команд архитектуры ARM. Результатом выполнения работы станет программа на ассемблере.

**Задание на лабораторную работу:**

**Вариант 8**

Деление целых чисел без знака для получения целого числа без восстановления остатка с неподвижным делителем и сдвигом делимого.

Текстовое описание алгоритма

1. Исходное значение частичного остатка (ЧО) полагается равным старшим разрядам делимого.
2. Из ЧО вычитается делитель и анализируется знак остатка.
3. Если остаток положительный, то деление невозможно, формируется признак переполнения и процесс завершается, в противном случае ЧО восстанавливается путем прибавления делителя и деление продолжается.
4. Частичный остаток сдвигается на один разряд влево, а в освобождающийся при сдвиге младший разряд ЧО заносится очередная цифра делимого.
5. Из сдвинутого ЧО вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицательный.
6. Очередная цифра модуля частного равна единице, если результат операции (сложения или вычитания) положителен, и нулю, если он отрицателен.
7. Пункты 4–6 последовательно выполняются для получения всех цифр модуля частного.

Блок-схема алгоритма



Рисунок 1. Блок-схема алгоритма

Листинг 1. Текст программы

; 25/4 = 6

;размер делителя - 4

start: mov r0, #25 ; делимое

mov r1, #4 ; делитель

mov r9, #0 ; r9 - счетчик

;r8 - ответ

mov r1, r1, lsl #4 ; сдвигаем делитель чтобы вычитать из старших разрядов

mov r3, r0, lsl #1 ; сдвигаем делимое

sub r3, r3, r1 ; вычитаем

cmp r3, #0

bgt finish ; если больше ноля, то выходим

mov r8, r8, lsl #1 ; записываем в ответ 0

add r9, r9, #1 ; увеличиваем счётчик

mov r3, r3, lsl #1 ; сдвигаем ЧО

add r3, r3, r1 ; складываем ЧО и делитель

loop: cmp r3, #0 ; сравниваем ЧО с нулём

bllt clow ; ЧО <0

blgt chigh ; ЧО >0

add r9, r9, #1 ; увеличиваем счётчик

cmp r9, #4 ; проверяем счётчик на конец деления

bne loop ; зацикливаем

beq finish ; выходим

clow: mov r8, r8, lsl #1 ; подпрограмма если ЧО <0

mov r3, r3, lsl #1 ; добавляем в ответ 0 и сдвигаем ЧО

add r3, r3, r1 ; складываем ЧО и делитель

bx lr

chigh: mov r8, r8, lsl #1 ; подпрограмма если ЧО > 0

add r8, r8, #1 ; добавляем в овтет 1

mov r3, r3, lsl #1 ; сдвигаем ЧО

sub r3, r3, r1 ; вычитаем делитель из ЧО

bx lr

finish: .end

|  |  |
| --- | --- |
| Инструкции программы | Состояние памяти и регистров |
| mov r0, #25 | R0 : 25 |
| mov r1, #4 | R1 : 4 |
| mov r9, #0 | R9 : 0 |
| mov r1, r1, lsl #4 | R1 : 64 |
| mov r3, r0, lsl #1 | R3 : 50 |
| sub r3, r3, r1 | R3 : -14 |
| cmp r3, #0 | N: 1  Z: 0  C: 1  V: 1 |
| bgt finish |  |
| mov r8, r8, lsl #1 | R8 : 0 |
| add r9, r9, #1 | R9 : 1 |
| mov r3, r3, lsl #1 | R3 = -28 |
| add r3, r3, r1 | R3 : 36 |
| cmp r3, #0 | N: 0  Z: 0  C: 1  V: 0 |
| bllt clow |  |
| blgt chigh | R14: 4156  R15: 4188 |
| mov r8, r8, lsl #1 | R8 : 0 |
| add r8, r8, #1 | R8 : 1 |
| mov r3, r3, lsl #1 | R3 : 72 |
| sub r3, r3, r1 | R3 : 8 |
| bx lr | R15 : 4156 |
| add r9, r9, #1 | R9 : 2 |
| cmp r9, #4 | N: 1  Z: 0  C: 0  V: 0 |
| bne loop | R15 : 4144 |
| cmp r3, #0 | N: 0  Z: 0  C: 1  V: 0 |
| bllt clow |  |
| blgt chigh | R14: 4156  R15: 4188 |
| mov r8, r8, lsl #1 | R8 : 2 |
| add r8, r8, #1 | R8 : 3 |
| mov r3, r3, lsl #1 | R3 : 16 |
| sub r3, r3, r1 | R3 : -48 |
| bx lr | R15 : 4156 |
| add r9, r9, #1 | R9 : 3 |
| cmp r9, #4 | N: 1  Z: 0  C: 0  V: 0 |
| bne loop | R15 : 4144 |
| cmp r3, #0 | N: 1  Z: 0  C: 1  V: 0 |
| bllt clow | R14: 4152  R15: 4172 |
| mov r8, r8, lsl #1 | R8 : 6 |
| mov r3, r3, lsl #1 | R3 : -96 |
| add r3, r3, r1 | R3 : -32 |
| bx lr | R15: 4152 |
| blgt chigh |  |
| add r9, r9, #1 | R9 : 4 |
| cmp r9, #4 | N: 0  Z: 1  C: 1  V: 0 |
| bne loop |  |
| beq finish | R15: 4208 |
| .end |  |

**Вывод:** изучил набор команд ARM. Написал программу по заданному алгоритму из ЛР1.