Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

ОТЧЕТ

По лабораторная работа №1

**Особенности выполнения арифметических**

**и логических операций в RISC процессорах**

По дисциплине «ЭВМ и периферийные устройства»

Выполнили:

Проверили:

Рязань 20

**Задание (Вариант 9):**

1. Изучить примеры использования логических операций,

приведенные в программе LR\_2. Для ввода программы создайте новый файл

в существующем проекте.(Таблица 1)

Таблица 1 — задание 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 1 | .include "m328pdef.inc" | подключение m328pdef.inc |  |
| 2 | ldi r16, 0x04 | r16:=0x08 в r16 загрузить 816 | r16=0x08 |
| 3 | out sph, r16 | sph:=(r16) в sph переслать содержимое r16 | sph=0x08 |
| 4 | ldi r16, 0x5f | r16:=0xff в r16 загрузить 25516 | r16=0xff |
| 5 | out spl, r16 | spl:=(r16) в spl переслать содержимое r16 | spl=0xff |
| 6 | call pp1 | вызов прцедуры pp1 |  |
| 9 | m: call pp2 | вызов прцедуры pp2 |  |
| 10 | call pp3LSL | вызов прцедуры pp3LSL |  |
| 11 | call pp3LSR | вызов прцедуры pp3LSR |  |
| 12 | call pp3ROL | вызов прцедуры pp3ROL |  |
| 13 | call pp3ROR | вызов прцедуры pp3ROR |  |
| 14 | call pp3ASR | вызов прцедуры pp3ASR |  |
| 15 | jmp m | перейти по метке m |  |
| 16 | pp1: ldi r16, 0x0f | r16:=0x0f в r16 загрузить 0x0f | r16=0x0f |
| 17 | ldi r17 , 0x45 | r17:=0x45 в r17 загрузить 0x45 | r17=0x45 |
| 18 | ldi r18 , 0x67 | r18:=0x67 в r18 загрузить 0x67 | r18=0x67 |
| 19 | ldi r19 , 0x89 | r19:=0x89 в r16 загрузить 0x89 | r19=0x89 |
| 20 | ldi r20 , 0x18 | r20:=0x18 в r16 загрузить 0x18 | r20=0x18 |
| Продолжение таблицы 1 — задание 1 | | | |
| 21 | and r17, r16 | r17:=r17&r16 логическое произведение к17 и к16 | r17=0x05 |
| 22 | andi r17,0b00000010 | r17:=r17&0b00000010 логическое произведение к17 и 0b00000010 | r17=0x00 |
| 23 | or r18, r16 | r18:=r18Vr16 логическая сумма r18 и r16 | r18=0x6f |
| 24 | tst r16 | r16 =r16&r16 Проверка на ноль либо отрицательность |  |
| 25 | eor r19, r16 | r19:=r19eorr16 r19 Логическое исключающее ИЛИ r16 | r19=0x86 |
| 26 | eor r19, r19 | r19:=r19eorr19 r19 Логическое исключающее ИЛИ r19 | r19=0x00 |
| 27 | mov r17, r20 | r17:=r20 в r17 переслать содержимое r20 | r17=0x18 |
| 28 | and r17, r16 | r17:=r17&r16 логическое произведение к17 и к16 | r17=0x08 |
| 29 | swap r16 | r16(3..0) = r16(7..4), r16(7..4) =r16(3..0) Перестановка тетрад | r16=0xf0 |
| 30 | and r18, r16 | r18:=r18&r16 логическое произведение к18 и к16 | r18=0x60 |
| 31 | or r17, r18 | r17:=r17Vr18 логическая сумма r17 и r18 | r17=0x68 |
| 32 | bst r16, 7 | T = r16(7) Сохранить бит из регистра в T | T=1 |
| 33 | bld r20, 2 | r20(2) = T Загрузить бит из T в регистр | r20=0x1c |
| 34 | sei | I = 1 Установить флаг прерываний | I=1 |
| 35 | cli | I = 0 Очистить флаг прерываний | I=0 |
| 36 | ret | выход из подпрограммы pp1 |  |

1. Написать подпрограмму формирования признака z=1, если введенное в регистр r5 произвольное двоичное число соответствует заданной маске. Приведите результаты тестирования программы при пересылке в r5 различных чисел

Для чисел 0b01010100, 0b11111111 и 0b01011111: (Таблица 2)

Таблица 2 — задание 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| итерация | № | Команда | Действие | Результат |
| 1 | 7 | ldi r16 , 0b01010100 | r16:=0b01010100 в r16 загрузить 0b01010100 | r16=0x54 |
|  | 8 | mov r5,r16 | r5:=r16 в r5 переслать содержимое r16 | r5=0x54 |
|  | 37 | pp2: |  |  |
|  | 38 | mov r17, r5 | r17:=r5 в r1 переслать содержимое r5 | r17=0x54 |
|  | 39 | andi r17, 0b00001111 | r17:=r17&0b00001111 логическое произведение к1 и 0b00001111 | r17=0x04 |
|  | 40 | ldi r16,0b01010000 | r16:=0b01010000 в r16 загрузить 0b01010000 | r16=0x50 |
|  | 41 | or r16, r17 | r16:=r16Vr17 логическая сумма r16 и r17 | r16=0x54 |
|  | 42 | eor r5, r16 | r5:=r5eorr16 r5 Логическое исключающее ИЛИ r16 | r5=0x00 Z=1 |
|  | 43 | ret |  |  |
| 2 | 7 | ldi r16 , 0b11111111 | r16:=0b11111111 в r16 загрузить 0b11111111 | r16=0xff |
|  | 8 | mov r5,r16 | r5:=r16 в r5 переслать содержимое r16 | r5=0xff |
|  | 37 | pp2: |  |  |
|  | 38 | mov r17, r5 | r17:=r5 в r1 переслать содержимое r5 | r17=0xff |
|  | 39 | andi r17, 0b00001111 | r17:=r17&0b00001111 логическое произведение к1 и 0b00001111 | r17=0x0f |
|  | 40 | ldi r16,0b01010000 | r16:=0b01010000 в r16 загрузить 0b01010000 | r16=0x50 |
| Продолжение таблицы 2 — задание 2 | | | | |
|  | 41 | or r16, r17 | r16:=r16Vr17 логическая сумма r16 и r17 | r16=0x5f |
|  | 42 | eor r5, r16 | r5:=r5eorr16 r5 Логическое исключающее ИЛИ r16 | r5=0xa0 Z=0 |
|  | 43 | ret |  |  |
| 3 | 7 | ldi r16 , 0b01011111 | r16:=0b01011111 в r16 загрузить 0b01011111 | r16=0x5f |
|  | 8 | mov r5,r16 | r5:=r16 в r5 переслать содержимое r16 | r5=0x5f |
|  | 37 | pp2: |  |  |
|  | 38 | mov r17, r5 | r17:=r5 в r1 переслать содержимое r5 | r17=0x5f |
|  | 39 | andi r17, 0b00001111 | r17:=r17&0b00001111 логическое произведение к1 и 0b00001111 | r17=0x0f |
|  | 40 | ldi r16,0b01010000 | r16:=0b01010000 в r16 загрузить 0b01010000 | r16=0x50 |
|  | 41 | or r16, r17 | r16:=r16Vr17 логическая сумма r16 и r17 | r16=0x5f |
|  | 42 | eor r5, r16 | r5:=r5eorr16 r5 Логическое исключающее ИЛИ r16 | r5=0x00 Z=1 |
|  | 43 | ret |  |  |

Составить подпрограмму pp3, позволяющую исследовать

выполнение операций сдвигов. (Таблица 3)

Таблица 3 — задание 3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| итерация | № | Команда | Действие | Рез. | 0b |
|  | 7 | ldi r16 , 0b01010100 | r16:=0b01010100 в r16 загрузить 0b01010100 | r16=0x5f |  |
|  | 8 | mov r5,r16 | r5:=r16 в r5 переслать содержимое r16 | r5=0x5f | 01011111 |
| LSL |  |  |  |  |  |
| 1 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0xbe | 10111110 |
| 2 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0x7c | 01111100 |
| 3 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0xf8 | 11111000 |
| 4 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0xf0 | 11110000 |
| 5 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0xe0 | 11100000 |
| 6 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0xc0 | 11000000 |
| 7 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0x80 | 10000000 |
| 8 | 45 | LSL r5 | Логический сдвиг влево r5 | r5=0x00 | 00000000 |
| LSR |  |  |  |  |  |
| итерация | № | Команда | Действие | Результат |  |
| 1 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x2f | 00101111 |
| 2 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x17 | 00010111 |
| 3 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x0b | 00001011 |
| 4 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x05 | 00000101 |
| 5 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x02 | 00000010 |
| 6 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x01 | 00000001 |
| 7 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x00 | 00000000 |
| 8 | 48 | LSR r5 | Логический сдвиг вправо r5 | r5=0x00 | 00000000 |
| Продолжение таблицы 3 — задание 3 | | | | | |
| ROL |  |  |  |  |  |
| итерация | № | Команда | Действие | Результат |  |
| 1 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0xbe | 10111110 |
| 2 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0x7c | 01111100 |
| 3 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0xf9 | 11111001 |
| 4 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0xf2 | 11110010 |
| 5 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0xe5 | 11100101 |
| 6 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0xcb | 11001011 |
| 7 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0x97 | 10010111 |
| 8 | 51 | ROL r5 | Циклический сдвиг влево через C r5 | r5=0x2f | 00101111 |
| ROR |  |  |  |  |  |
| итерация | № | Команда | Действие | Результат |  |
| 1 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0x2f | 00101111 |
| 2 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0x97 | 10010111 |
| 3 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0xcb | 11001011 |
| 4 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0xe5 | 11100101 |
| 5 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0xf2 | 11110010 |
| 6 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0xf9 | 11111001 |
| 7 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0x7c | 01111100 |
| 8 | 54 | ROR r5 | Циклический сдвиг вправо через C r5 | r5=0xbe | 10111110 |
| ASR |  |  |  |  |  |
| итерация | № | Команда | Действие | Результат |  |
| 1 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x2f | 00101111 |
| Завершение таблицы 3 — задание 3 | | | | | |
| 2 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x17 | 00010111 |
| 3 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x0b | 00001011 |
| 4 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x05 | 00000101 |
| 5 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x02 | 00000010 |
| 6 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x01 | 00000001 |
| 7 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x00 | 00000000 |
| 8 | 57 | ASR r5 | Арифметический сдвиг вправо C r5 | r5=0x00 | 00000000 |

Вывод:

Логический сдвиг влево сдвигает все биты влево, младший приравнивает к 0.

Логический сдвиг вправо сдвигает все биты вправо, старший приравнивает к 0.

Циклический сдвиг влево через C сдвигает все биты влево, младший приравнивает к прошлому значению старшего.

Циклический сдвиг вправо через C сдвигает все биты вправо, старший приравнивает к прошлому значению младшего.

Арифметический сдвиг вправо C сдвигает все биты вправо, сохраняя старший бит.

1. Исследование выполнения операций суммирования

Изучить примеры использования арифметических операций, приведенные в программе LR\_3. Для ввода программы создайте новый файл в существующем проекте.

Таблица 4 — задание 4 прямой код начало таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 1 | .include "m328pdef.inc" | подключение m328pdef.inc |  |
| 2 | ldi r16, 0x04 | r16:=0x04 в r16 загрузить 4 | r16=0x04 |
| 3 | out sph, r16 | sph:=(r16) в sph переслать содержимое r16 | sph=0x04 |
| 4 | ldi r16, 0x5f | r16:=0x5f в r16 загрузить 95 | r16=0x5f |
| 5 | out spl, r16 | spl:=(r16) в spl переслать содержимое r16 | spl=0x5f |
| 6 | call pp1 | вызов прцедуры pp1 |  |
| 7 | call pp2 | вызов прцедуры pp2 |  |
| 8 | call pp3 | вызов прцедуры pp3 |  |
|  | call pp4 | вызов прцедуры pp4 |  |
|  | call pp5 | вызов прцедуры pp5 |  |
| 9 | m: jmp m | перейти по метке m |  |
| 10 | pp1: ldi r16 , 0b00100111 | r16:=0x27 в r16 загрузить 39 | r16=0x27 |
| 11 | ldi r17 , 0b01101101 | r17:=0x6d в r17 загрузить 109 | r17=0x6d |
| 12 | ldi r18 , 0b11000110 | r18:=0xc6 в r18 загрузить 198 | r18=0xc6 |
| 13 | ldi r19 , 0b11101111 | r19:=0xef в r19 загрузить 239 | r19=0xef |
| 14 | add r16, r17 | r16:=r16+r17 r16 загрузить сумму r16 и к17 | r16=0x94 |
| 15 | add r17, r18 | r17:=r17+r18 r17 загрузить сумму r17 и к18 | r17=0x33 |
| 16 | add r18, r19 | r18:=r18+r19 r18 загрузить сумму r18 и к19 | r18=0xb5 |
| Таблица 4 — задание 4 прямой код завершение таблицы | | | |
| 17 | ret | выход из подпрограммы pp1 |  |

Таблица 5 — задание 4 дополнительный код

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 10 | pp1: ldi r16 , 0b11011001 | r16:=0x27 в r16 загрузить -39 | r16=0xde |
| 11 | ldi r17 , 0b10010011 | r17:=0x6d в r17 загрузить -109 | r17=0x93 |
| 12 | ldi r18 , 0b11000110 | r18:=0xc6 в r18 загрузить 198 | r18=0xc6 |
| 13 | ldi r19 , 0b11101111 | r19:=0xef в r19 загрузить 239 | r16=0xef |
| 14 | add r16, r17 | r16:=r16+r17 r16 загрузить сумму r16 и к17 | r16=0x6c |
| 15 | add r17, r18 | r17:=r17+r18 r17 загрузить сумму r17 и к18 | r17=0x59 |
| 16 | add r18, r19 | r18:=r18+r19 r18 загрузить сумму r18 и к19 | r18=0xb5 |

Проверка на калькуляторе:

Прямой код:

39+109=148=0x94 верно

109+198=307=0x133 верно, учитывая что хранится только младший байт результата

198+239=437=0x1b5 верно, учитывая что хранится только младший байт результата

Дополнительный код:

-39-109=-148=0x16c верно, учитывая что хранится только младший байт результата

-109+198=89=0x59 верно

198+239=437=0x1b5 верно, учитывая что хранится только младший байт результата

1. Вычисление среднего значения двух чисел

Выполнить подпрограмму рр2 по шагам.

Таблица 6 — задание 5 начало таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 18 | pp2: ldi r16 , 250 | r16:=0xfa в r16 загрузить 250 | r16=0xfa |
| 19 | ldi r17 , 200 | r17:=0xc8 в r17 загрузить 200 | r17=0xc8 |
| 20 | mov r0, r16 | r0:=(r16) в r0 переслать содержимое r16 | r0=0xfa |
| 21 | add r0, r17 | r0:=r0+r17 r0 загрузить сумму r0 и к17 | r0=0xc2 |
| 22 | ror r0 | Циклический сдвиг вправо через C r0 | r0=0xe1 |
| 23 | ret | выход из подпрограммы pp2 |  |

1. Вычисление среднего значения нескольких чисел

Проверить работу программы для двух вариантов представления чисел - числа положительные, представлены в прямом коде и числа со знаком, представлены в дополнительном коде.

Таблица 7 — задание 6 прямой код начало таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 24 | pp3: ldi r16 , 0b00100111 | r16:=0x27 в r16 загрузить 39 | r16=0x27 |
| 25 | ldi r17 , 0b01101101 | r17:=0x6d в r17 загрузить 109 | r17=0x6d |
| 26 | ldi r18 , 0b11000110 | r18:=0xc6 в r18 загрузить 198 | r18=0xc6 |
| 27 | ldi r19 , 0b11101111 | r19:=0xef в r19 загрузить 239 | r16=0xef |
| 28 | eor r1, r1 | r1:=r1eorr1 r1 Логическое исключающее ИЛИ r1 | r1=0x00 |
| 29 | eor r2, r2 | r2:=r2eorr2 r2 Логическое исключающее ИЛИ r2 | r2=0x00 |
| 30 | mov r0, r16 | r0:=(r16) в r0 переслать содержимое r16 | r0=0x27 |
| 31 | add r0, r17 | r0:=r0+r17 r0 загрузить сумму r0 и к17 | r0=0x94 |
| 32 | adc r1 ,r2 | r1:=r1+r2+c суммирование с переносом | r0=0x94 |
| 33 | add r0, r18 | r0:=r0+r18 r0 загрузить сумму r0 и к18 | r0=0x5a |
| 34 | adc r1 ,r2 | r1:=r1+r2+c суммирование с переносом | r1=0x01 |
| Таблица 7 — задание 6 прямой код завершение таблицы | | | |
| 35 | add r0, r19 | r0:=r0+r19 r0 загрузить сумму r0 и к19 | r0=0x49 |
| 36 | adc r1 ,r2 | r1:=r1+r2+c суммирование с переносом | r1=0x02 |
| 37 | ror r1 | Циклический сдвиг вправо через C r1 | r1=0x01 |
| 38 | ror r0 | Циклический сдвиг вправо через C r0 | r0=0x24 |
| 39 | ror r1 | Циклический сдвиг вправо через C r1 | r1=0x80 |
| 40 | ror r0 | Циклический сдвиг вправо через C r0 | r0=0x92 |
| 41 | ret | выход из подпрограммы pp3 |  |

Таблица 8 — задание 6 дополнительный код

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 24 | pp3: ldi r16 , 0b11011001 | r16:=0x27 в r16 загрузить -39 | r16=0xd9 |
| 25 | ldi r17 , 0b10010011 | r17:=0x6d в r17 загрузить -109 | r17=0x93 |
| 26 | ldi r18 , 0b11000110 | r18:=0xc6 в r18 загрузить 198 | r18=0xc6 |
| 27 | ldi r19 , 0b11101111 | r19:=0xef в r19 загрузить 239 | r16=0xef |
| 39 | ror r1 | Циклический сдвиг вправо через C r1 | r1=0x80 |
| 40 | ror r0 | Циклический сдвиг вправо через C r0 | r0=0xc8 |

Проверка на калькуляторе:

Прямой код:

(39+109+189+239)/4=146=0x92 верно

Дополнительный код:

(-39-109+198+239)/4=72=0x48 верно

Расчёт среднеарифметического работает правильно, даже при том, что (39+109+189+239)>0xff.

1. Умножение положительных чисел

Составьте подпрограмму для исследования команды умножения mul, которая

перемножает однобайтные положительные сомножители, размещенные в

регистрах, указанных в команде.

Таблица 9 — задание 7 начало таблицы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Команда | Действие | Результат |
| 42 | pp4: ldi r16 , 0b00100111 | r16:=0x27 в r16 загрузить 39 | r16=0x27 |
| 43 | ldi r17 , 0b01101101 | r17:=0x6d в r17 загрузить 109 | r17=0x6d |
| 44 | ldi r18 , 0b11000110 | r18:=0xc6 в r18 загрузить 198 | r18=0xc6 |
| 45 | ldi r19 , 0b11101111 | r19:=0xef в r19 загрузить 239 | r16=0xef |
| 46 | mul r16, r17 | r16:=r16\*r17 r16 загрузить произведение r16 и к17 | r0=0x9b r1=0x10 |
| 47 | mul r17, r18 | r17:=r17\*r18 r17 загрузить произведение r17 и к18 | r0=0x4e r1=0x54 |
| 48 | mul r18, r19 | r18:=r18\*r19 r18 загрузить произведение r18 и к19 | r0=0xda r1=0xb8 |
| 49 | ret | выход из подпрограммы pp4 |  |

1. Умножение чисел со знаком

Составить подпрограмму для исследования команды умножения muls.

Таблица 10 — задание 8

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | muls A, B | A\*B |
| 39 | 109 | r0=0x9b r1=0x10 | 4251 |
| -39 | -109 | r0=0x9b r1=0x10 | 4251 |
| -39 | 109 | r0=0x65 r1=0xef | -4251 |

Умножение со знаком работает корректно, результат сохраняется в r0 - младшицй байт произведения и в r1 - старший.

Вывод: ознакомились с особенностями выполнения арифметических

и логических операций в RISC процессорах.