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 |
| ПРОЕКТ АРХИТЕКТУРЫ НАБОРА КОМАНД И МИКРОАРХИТЕКТУРЫ ПРОСТЕЙШЕГО 8-БИТНОГО ПРОЦЕССОРА |
| по курсу: ОРГАНИЗАЦИЯ ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2022

**Цель работы:** разработать модель 8-битного процессора

**Задание на лабораторную работу**

**Вариант 7:**

Вид архитектуры: Гарвардская (команды и данные в разных ЗУ)

Тип данных: беззнаковый

Операции в АЛУ:

Деление с остатком

lsr

not

Пересылка: mov

Переход по условию, в соответствии с операцией сравнения: <

Формат команды: КОП АО1 АО2

Способы адресации:

1. Косвенная адресация по смещению
2. Адресация с автоувеличением (простая косвенная с автоувеличением)

**Разработка системы команд**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код | Команда | Свойство | Кол-во оп. | Функция АЛУ | Сигналы УУ | Флаги |
| 0000 | NOT | Логическое отрицание | 1 | Логическое отрицание | CC – 4 бита код команды на АЛУ  DR – 1 выдать данные в ШД данных  RO – команда АЛУ записать операнд в аккумулятор  Ready – команда от АЛУ о выполнении операции |  |
| 0001 | LSR | Сдвиг вправо | 2 | Сдвиг вправо | CC – 4 бита код команды на АЛУ  DR – 1 выдать данные в ШД данных  RO – команда АЛУ записать операнд в аккумулятор  Ready – команда от АЛУ о выполнении операции | C |
| 0010 | DIV | Деление с остатком | 2 | Деление с остатком | CC – 4 бита код команды на АЛУ  DR –выдать данные в ШД данных  Dw – записать в память данных из ШД  RO – команда АЛУ записать операнд в аккумулятор  Rw – записать остаток в РОН  RN- номер регистра  Ready – команда от АЛУ о выполнении операции |  |
| 0011 | MOVB | Пересылка байта | 2 |  | RR – прочитать из РОН  RN- номер регистра  DR – выдать данные в ШД данных  DA- адрес в памяти данных |  |
| 0111 | MOVW | Пересылка слова | 2 |  | RR – прочитать из РОН  RN- номер регистра  DR – выдать данные в ШД данных  DA- адрес в памяти данных |  |
| 1111 | MOVD | Пересылка двойного слова | 2 |  | RR – прочитать из РОН  RN- номер регистра  DR – выдать данные в ШД данных  DA- адрес в памяти данных |  |
| 0100 | BLLSSU | Переход если меньше | 1 |  | RR – прочитать из РОН  RN- номер регистра  Увеличить PC на значение из регистра | С |

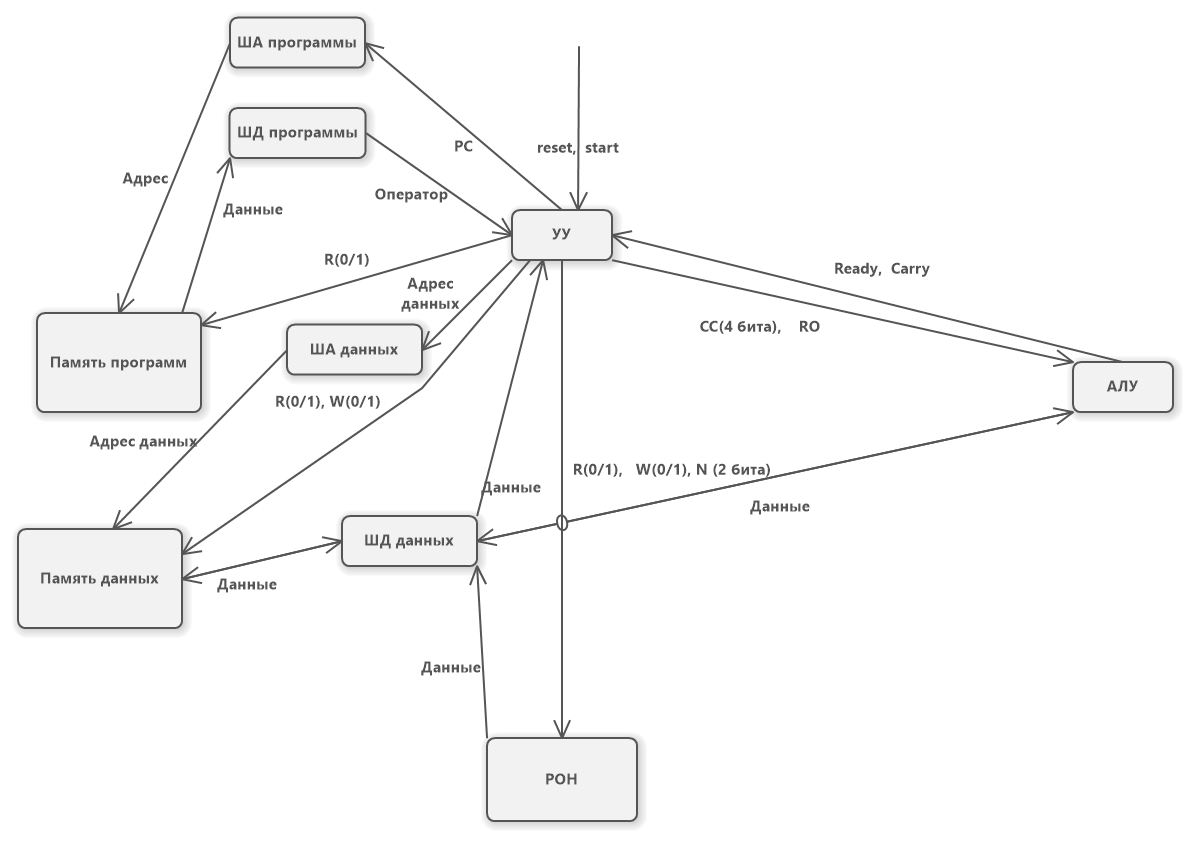


Рисунок 1. Детализированная схема архитектуры

**Примеры кодировки режимов адресации**

|  |  |  |
| --- | --- | --- |
| Мнемокод | Машинный код | Комментарий |
| MOVB (R0)+,(R0)+ | 00000011  10000000  10000000 | Начальное значение R0 – 00000000  Команда переместит байт по адресу 0h в адрес 1h |
| MOVD (R0)+,(R0)+ | 00001111  10000000  10000000 | Начальное значение R0 – 00000000  Команда переместит 4 байта по адресу 0h-3h в адрес 4h-7h |
| MOVB (R0)+, 5 (R0) | 00000011  10000000  00000101 | Начальное значение R0 – 00000000  Команда переместит байт по адресу 0h в адрес 6h |

КОП, бит адресации, номер регистра, смещение.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| КОП | | | | | | | |  | Адресация | | |  |  |  |  |  |  | Адресация | | |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |  | Регистр | | |  |  |  |  |  |  | Регистр | |  | Смещение | | | |

**Таблица входов/выходов**

**АЛУ**

|  |  |
| --- | --- |
| Входы | Выходы |
| СС – код команды на выполнения от УУ | Ready – сигнал выполнения команды |
| RO – сигнал на запись операнда из ШД во внутренний аккумулятор | Carry – флаг результата операции |

**УУ**

|  |  |
| --- | --- |
| Входы | Выходы |
| Reset – сброс PC | PR– чтение из памяти программ |
| Start – запуск программы | PC – программный счетчик в ША программы |
| PO – оператор из памяти программы | DR\DW – чтение\запись в память данных |
| D – данные из ШД памяти данных | DA- адрес в память данных |
| Ready – сигнал выполнения команды от АЛУ | СС – код команды на выполнения для АЛУ |
| Carry – флаг результата операции от АЛУ | RO – сигнал на запись операнда из ШД во внутренний аккумулятор АЛУ |
| RR\RW – чтение\запись в РОН |  |
| RN – номер регистра |  |

**РОН**

|  |  |
| --- | --- |
| Входы | Выходы |
| RR\RW – чтение\запись в РОН | D – данные в ШД памяти данных |
| RN – номер регистра |  |

**Память программ**

|  |  |
| --- | --- |
| Входы | Выходы |
| PR– чтение из памяти программ | P – данные из памяти |
| A – адрес из ША |  |

**Память данных**

|  |  |
| --- | --- |
| Входы | Выходы |
| DR\DW – чтение\запись в память данных | D – данные из памяти |
| A – адрес из ША |  |
| D – данные из памяти |  |

**Детальное текстовое описание:**

При подаче сигнала start УУ вычисляет адрес команды, передает его на ША памяти программ (ПП), выдает сигнал на чтение из ПП. Через ШД ПП получает команду, четыре младших бита которой передает в АЛУ. Затем инкрементирует PC. Если у оператора имеются операнды, то считывается в начале их тип адресации. Адресация записывается в старшем бите операнда. 0 – косвенная адресация по смещению. 1 – адресация с автоувеличением.

Косвенная адресация по смещению предполагает хранение в операнде номера РОН и смещения в виде константы. Так как регистров всего 4, значит два старших бита оставляются под номера регистров(не считая бит адресации), а остальные 5 для значения смещения.

Адресация с автоувеличением после выборки операнда значение регистра хранящего адрес операнда автоматически увеличится на 1, 2, 4 и т.д. в зависимости от команды. Работает с командой пересылки.

Когда адресация определена, УУ пересылает в РОН адрес регистра и команду на чтение. РОН выдает значение на ШД, а АЛУ получает команду на запоминание операнда.

После выполнения команды АЛУ отсылает сигнал Ready на УУ и выдает значение на ШД. А УУ запрашивает новую команду из памяти программ.

**Вывод:** разработал модель 8-битного процессора, а также детализировал структуру процессора.