**Цель работы**

Изучить организацию ввода и вывода информации через параллельные порты микроконтроллера, особенности работы с отдельными линиями параллельных портов, специфику использования отдельных команд микроконтроллера и его режимов адресации.

**Задание**

В зависимости от варианта, микроконтроллер осуществляет ввод (*ВВ*) либо вывод (*ВЫВ*) информации.

Данные представляют собой десятичные цифры, записанные либо в упакованном (*УП*), либо в распакованном (*РАСП*) формате. В первом случае в каждом байте хранятся по две десятичные цифры, старшая занимает левую тетраду, а младшая — правую. Во втором случае каждая цифра занимает лишь младшую тетраду байта, старшая тетрада при обмене не должна меняться.

Начальный адрес массива, из которого выводятся или в который вводятся данные, либо хранится в регистре *R0* текущего банка данных, либо имеет фиксированное значение, указанное в таблице вариантов. Длина массива, задаваемая в цифрах, либо хранится в регистре *R1*, либо равна первой цифре массива (*ПЦ*), либо конец обмена определяется сигналом низкого уровня на входе *P1.7*. В первом случае цифра, определяющая длину массива, входит в состав массива и должна передаваться вместе с массивом. Обмен осуществляется параллельным (*ПР*) по каналам *P1.3...P1.0* либо последовательным по каналу *P1.0* кодом. Последовательная передача может осуществляться старшими (*ПССТ*) либо младшими (*ПСМЛ*) битами вперед.

Инициатором обмена выступает микроконтроллер (*МК*) либо внешнее устройство (*ВУ*). В первом случае каждая операция обмена начинается с выдачи микроконтроллером сигнала запроса к *ВУ*. Обмен осуществляется после приема от *ВУ* сигнала подтверждения, при поступлении которого *МК* производит обмен, снимает запрос, ждет снятия сигнала подтверждения со стороны *ВУ* и затем продолжает работу.

Во втором случае прежде, чем начать обмен *МК* ожидает сигнала запроса от *ВУ*, затем выставляет сигнал готовности к обмену, производит обмен, снимает сигнал готовности, ждет снятия запроса от *ВУ*, после чего продолжает работу.

Обмен управляющими сигналами между *МК* и *ВУ* осуществляется при каждой передаче. *ВУ* передает сигналы в *МК* (подтверждение или запрос) по линии *Р1.4*. *МК* передает сигналы к *ВУ* (запрос или готовность) по линии *Р1.5*. Активное значение сигнала указывается в варианте задания (*Н* – высокий, *L –* низкий).

После завершения передачи массива управление передается на начало программы.

Таблица 1 – Задание варианта 23.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ варианта** | **Вид числа** | **ВВ/ ВЫВ** | **Вид обмена** | **Начальный адрес** | **Длина массива** | **Инициатор** | **Уровни сигналов** | |
| **МК** | **ВУ** |
| 23 | РАСП | ВВ | ПСМЛ | 10H | R1 | МК | L | H |

Согласно варианту, микроконтроллер осуществляет ввод распакованных чисел параллельным кодом по каналам *P1.3...Р1.0*. Начальный адрес массива – *20H*. Длина массива – его первая цифра. Инициатор обмена – *МК*. Запрос от *МК –* сигнал низкого уровня по каналу *Р1.4*. Подтверждение от *ВУ –* сигнал высокого уровня по каналу *Р1.5*.

**Решение**

Программа реализации данного варианта задания с необходимыми комментариями приведена ниже.

**ORG 000H**

**SJMP START ; переход на начало программы**

**ORG 030H**

**START:**

**MOV R1, #5 ; R1 - длина массива**

**MOV P1, #00110001b ; Р1.5 - высокий, Р1.4 и P1.0 - на ввод**

**MOV R0, #10H ; R0 — начальный адрес массива (i=0)**

**INDG: ACALL RESV ; переход на подпрограмму ввода цифры**

**XCH A, @R0 ; обмен значениями A и ячейки**

**ANL A, #0F0H ; оставляем старшую тетраду нетронутой**

**ORL A, @R0 ; записываем цифру массива в младшую тетраду**

**MOV @R0, A ; запись в ячейку памяти**

**INC R0 ; и переход к адресу след. ЯП массива**

**DJNZ R1, INDG ; если R1<>0, то на ввод следующей цифры**

**AJMP START ; зацикливание программы**

**; подпрограмма ввода цифры**

**RESV: MOV R2, #4 ; счетчик бит в цифре**

**CLR A ; очистка A**

**IN: CLR P1.5 ; P1.5 - низкий (запрос от МК)**

**JNB P1.4, $ ; ожидание подтверждения от ВУ**

**MOV C, P1.0 ; ввод бита в перенос**

**SETB P1.5 ; снятие запроса**

**JB P1.4, $ ; ожидание снятия подтверждения**

**RRC A ; A<7> = (введенный бит)**

**DJNZ R2, IN ; цикл ввода 4-х бит**

**SWAP A ; перемещаем цифру в мл. тетраду**

**RET ; возврат из подпрограммы**

**END**

**Результат работы программы**

Результат работы программы – ввод массива {9, 2, 5, 6, 7} в среде ProView32 представлен на рисунках 1 – 6.

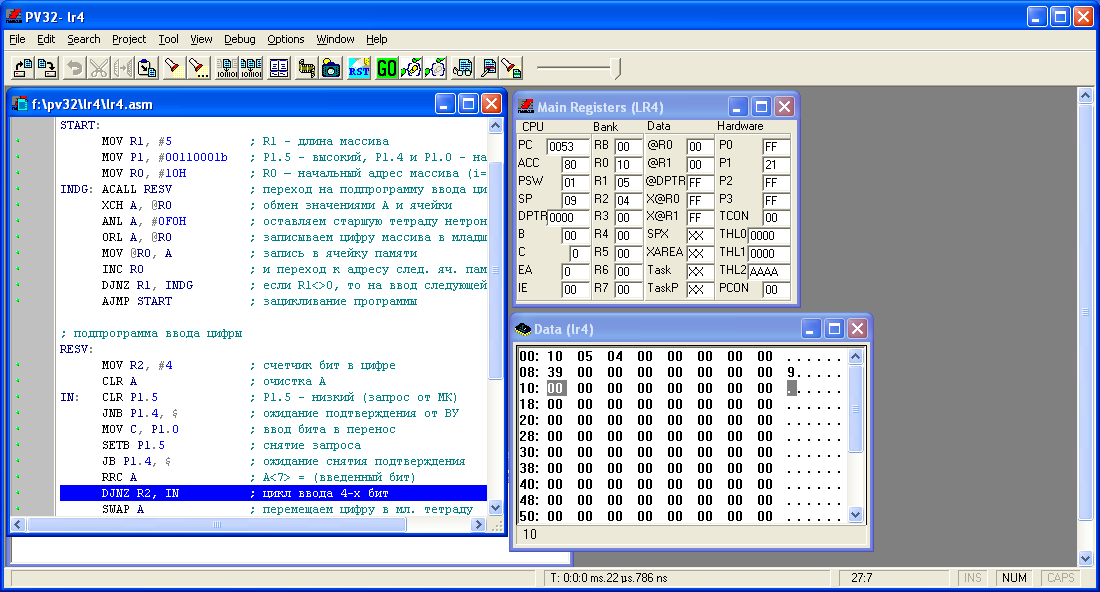


Рисунок 1 – Прием первого бита цифры

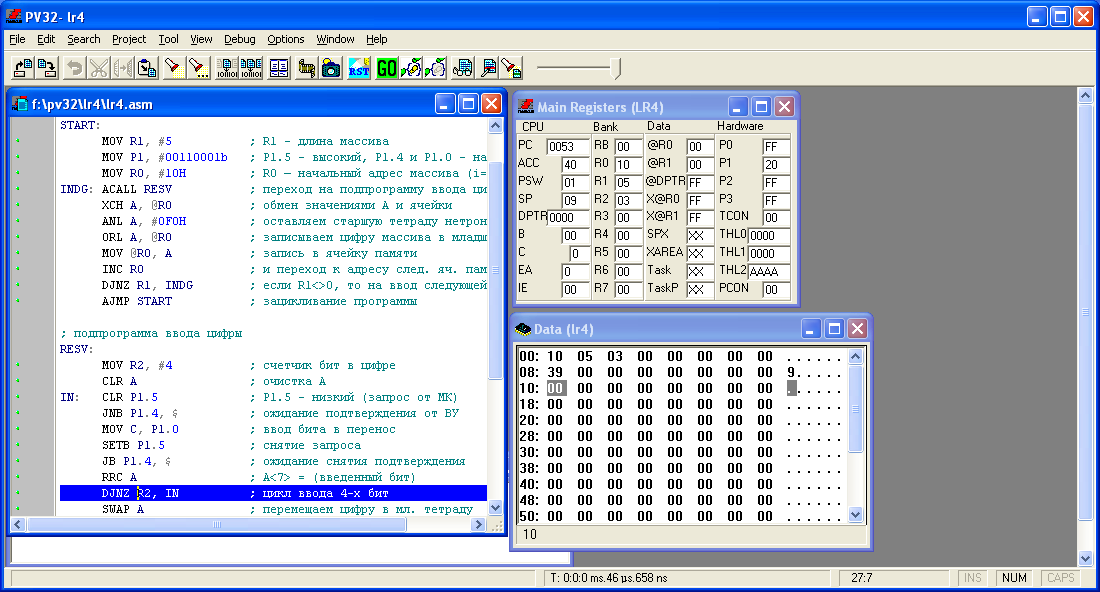


Рисунок 2 – Прием второго бита цифры

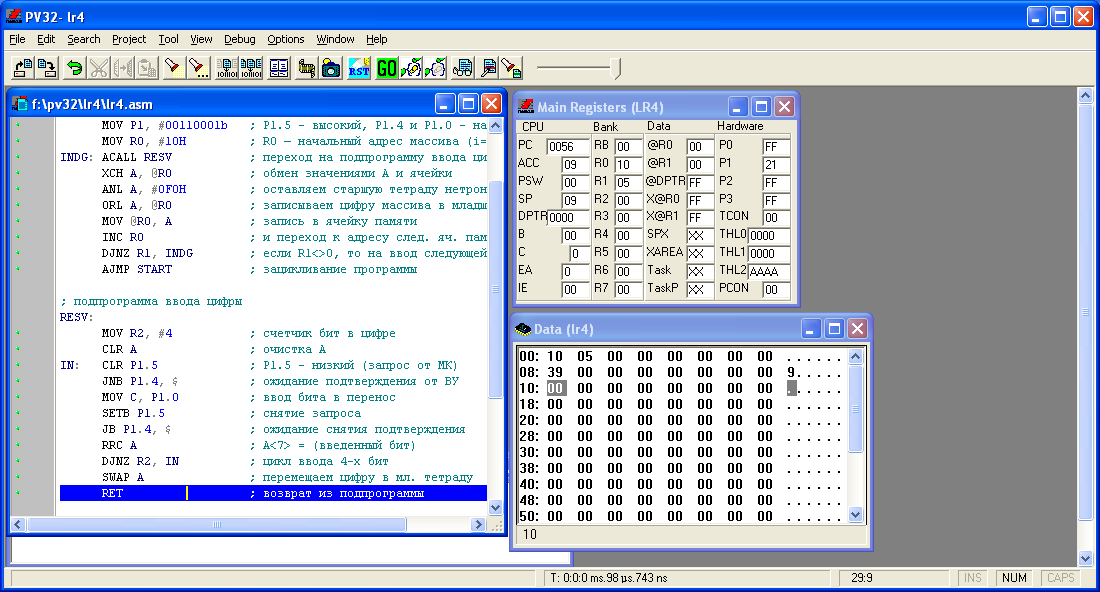


Рисунок 3 – Конец приема цифры и ее размещение в младшей тетраде аккумулятора

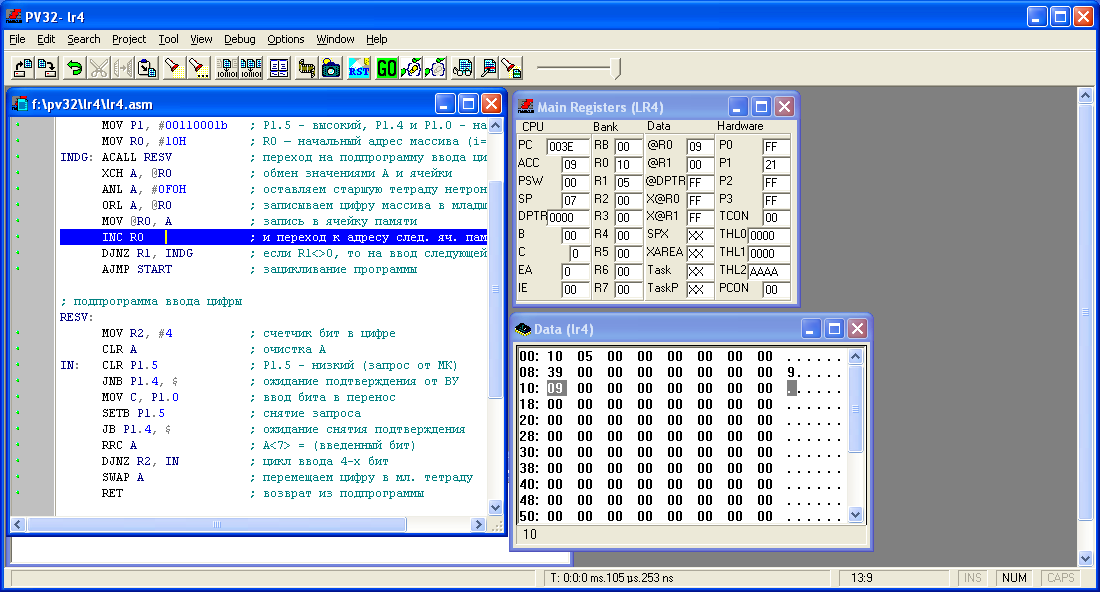


Рисунок 4 – Запись цифры в массив без изменения старшей тетрады байта

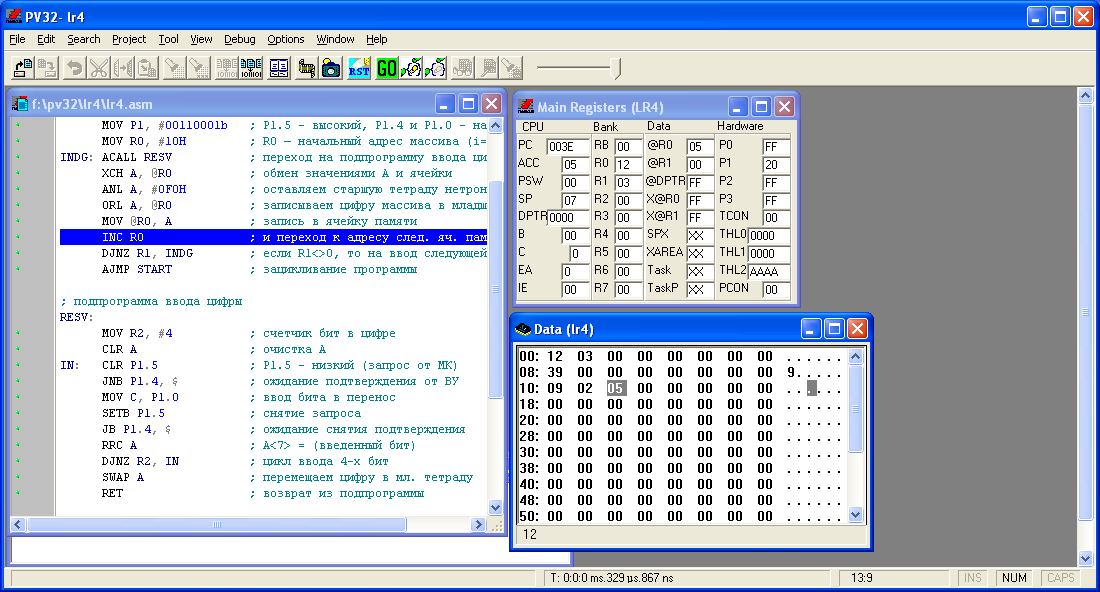


Рисунок 5 – Запись 3-ей принятой цифры в массив

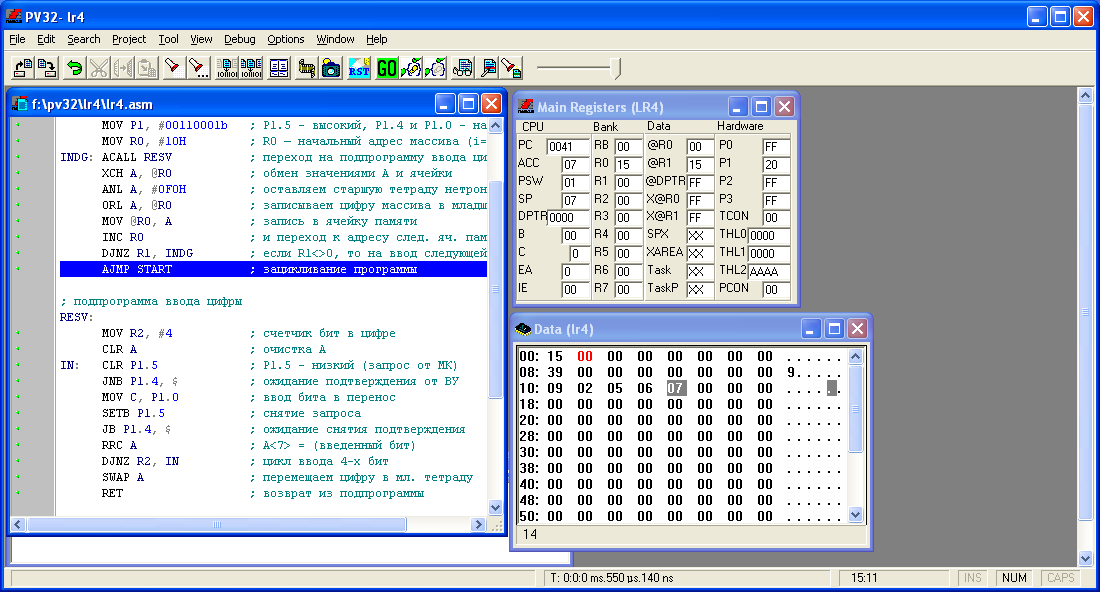


Рисунок 6 – Конец приема массива из 5 цифр

**Выводы**

В ходе лабораторной работы были изучены: организация ввода и вывода информации через параллельные порты микроконтроллера, особенности работы с отдельными линиями параллельных портов, специфика использования отдельных команд микроконтроллера и его режимов адресации.

В результате была разработана программа реализации заданного варианта по вводу массива.

Тестирование и отладка программы велись в среде разработки ProView. Снимки с экрана среды разработки, подтверждают полноту решения задачи и правильность работы программы.