

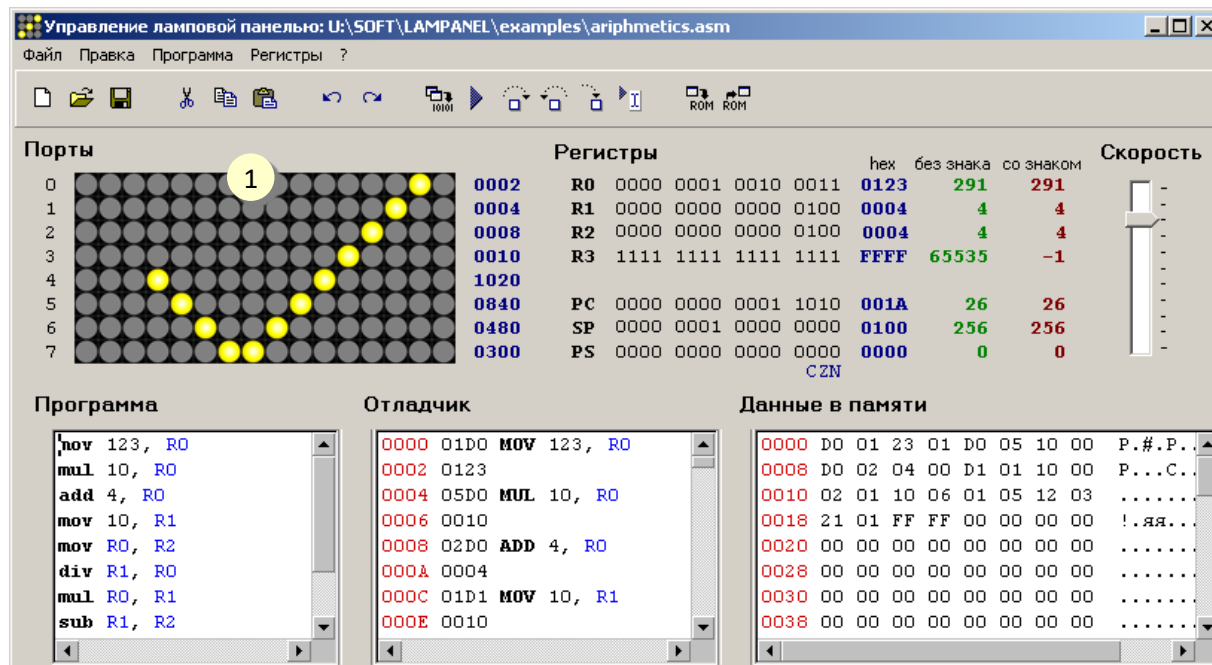
Практическая работа №13

Для выполнения этих работ используется учебный компьютер «ЛамПанель», который можно загрузить со страницы <http://kpolyakov.spb.ru/prog/lamp.htm>.

Процессор и устройства вывода

Возможности программы «ЛамПанель»

Наконец, мы подошли к самой интересной возможности программы «ЛамПанель» - управлению ламповой панелью. Ламповая панель (цифра 1 на рисунке) – это устройство вывода.



Обмен данными процессора и внешнего устройства происходит через *порты* – регистры контроллера внешнего устройства. У ламповой панели 8 портов, которые называются **P0**, **P1**, **P2**, **P3**, **P4**, **P5**, **P6** и **P7**. Каждый порт «отвечает» за одну строку лампочек, например, для того, чтобы «зажечь» всю верхнюю строку нужно записать в порт **P0** код $FFFF_{16}$ (все 16 бит – единичные). Для этого можно использовать, например, команды

```
MOV FFFF, R0
OUT R0, P0
```

К сожалению, записать число сразу в порт нельзя – сначала нужно записать его в регистр (в данном примере – в **R0**), а потом – из регистра в порт.

Для того, чтобы изменить второй сверху ряд лампочек, нужно записать новое значение в **P1** и т.д.; последний ряд управляется портом **P7**. Например, для того, чтобы все ряды лампочек горели одинаково, можно записать нужный код сначала в регистр:

```
MOV AAAA, R0
```

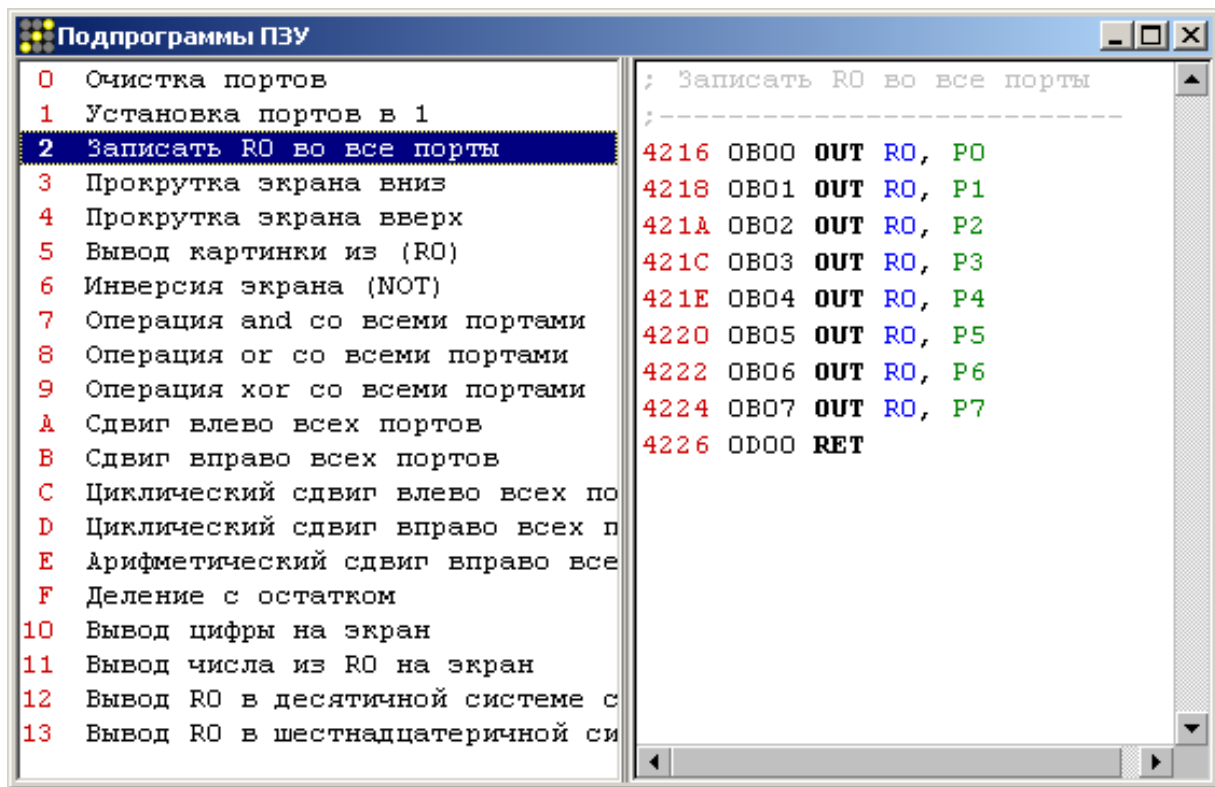
а затем из этого регистра – во все порты:

```
OUT R0, P0
OUT R0, P1
...
OUT R0, P7
```

Здесь многоточие обозначает аналогичные команды записи содержимого регистра **R0** в порты **P2...P6**. Однако вместо последней серии из 8 команд можно использовать всего одну:

```
SYSTEM 2
```

Эта команда вызывает системную процедуру с номером 2, находящуюся в ПЗУ компьютера. Для того, чтобы увидеть все процедуры, которые записаны в ПЗУ, нужно щелкнуть по кнопке  или выбрать пункт верхнего меню *Программа – Просмотр ПЗУ*. После этого появляется окно, в левой части которого перечислены все системные процедуры (с их номерами), а в правой части показывается код выбранной процедуры:



В этом списке есть много полезных процедур, в том числе

- 0 – очистка экрана (погасить все лампочки);
- 1 – зажечь все лампочки на панели;
- 3-4 – прокрутка изображения вниз и вверх;
- 6-9 – логические операции;
- A₁₆-E₁₆ – сдвиги битов;
- 12₁₆ – вывод числа, записанного в регистр R0, в десятичной системе счисления;
- 13₁₆ – вывод числа, записанного в регистр R0, в шестнадцатеричной системе счисления.

Обратите внимание, что номер системной процедуры задается в шестнадцатеричной системе счисления.

Рассмотрим еще одну задачу: вывести на экран рисунок, закодированный в виде шестнадцатеричных чисел (бит, равный единице, обозначает горящую лампочку). Для этого нужно сначала записать коды рисунка в память. Поскольку наш компьютер основан на архитектуре фон Неймана, в нем программа и данные находятся в одной области памяти. Поэтому данные можно записать с помощью специальной команды **DATA** после команды **STOP**:

```

...           ; здесь будет программа
STOP
M:            ; метка – начало блока данных
DATA AAAA    ; код первой строчки
DATA 5555
DATA AAAA
DATA 5555
DATA AAAA
DATA 5555
DATA AAAA
DATA 5555    ; код последней строчки

```

Для того, чтобы вывести этот рисунок на экран, нужно записать его адрес в регистр R0 и вызвать системную процедуру с номером 5:

```

MOV @M, R0    ; записать адрес метки M в регистр R0
SYSTEM 5      ; вывести на экран рисунок, адрес которого в R0
STOP
M:
DATA AAAA    ; код первой строчки
...

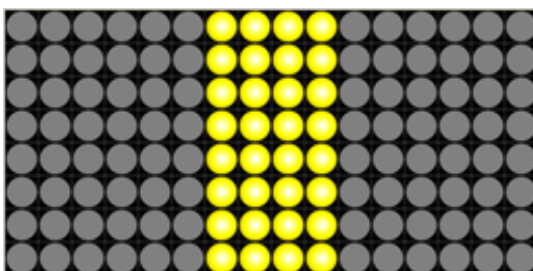
```

Задание на практическую работу

1. Запишите в таблицу минимальное и максимальное числа, которые можно вывести на ламповую панель, если использовать шестнадцатеричную систему:

минимальное число		максимальное число	
десятичное	шестнадцатеричное	десятичное	шестнадцатеричное

2. Составьте программу, после выполнения которой ламповая панель выглядит так:



Программа:

3. Как вы думаете, что выведет приведенная выше (в теоретической части) программа, которая вызывает системную процедуру с номером 5? Проверьте ваш ответ с помощью тренажёра.

Ответ:

4. Закодируйте изображение домика и выведите его на экран.

Программа:

5. Добавьте в предыдущую программу команды, которые сначала шифруют изображение, используя операцию «исключающее ИЛИ» с маской $BCA7_{16}$, а затем – восстанавливают исходное изображение. При изменении маски программа не должна изменяться. Изучите текст системной процедуры, которую вы используете.

Программа:

6. Напишите программу, которая делает «бегущую строку» из рисунка-домика.

Программа:

7. Напишите программу, которая организует «обратный отсчет» от 100 до 0, а затем выводит рисунок с домиком и останавливается.

Программа: