Национальный исследовательский университет «МЭИ»

Институт информатики и вычислительных технологий

Лабораторная работа №1 по курсу «ЭВМиПУ»

**Часть 1**

|  |  |
| --- | --- |
| Выполнил |  |
| Студент | Балашов С.А. |
| **Группа** | А-08-19 |

|  |  |
| --- | --- |
| Принял |  |
| Преподаватель | Карпов А.В. |
| Дата |  |

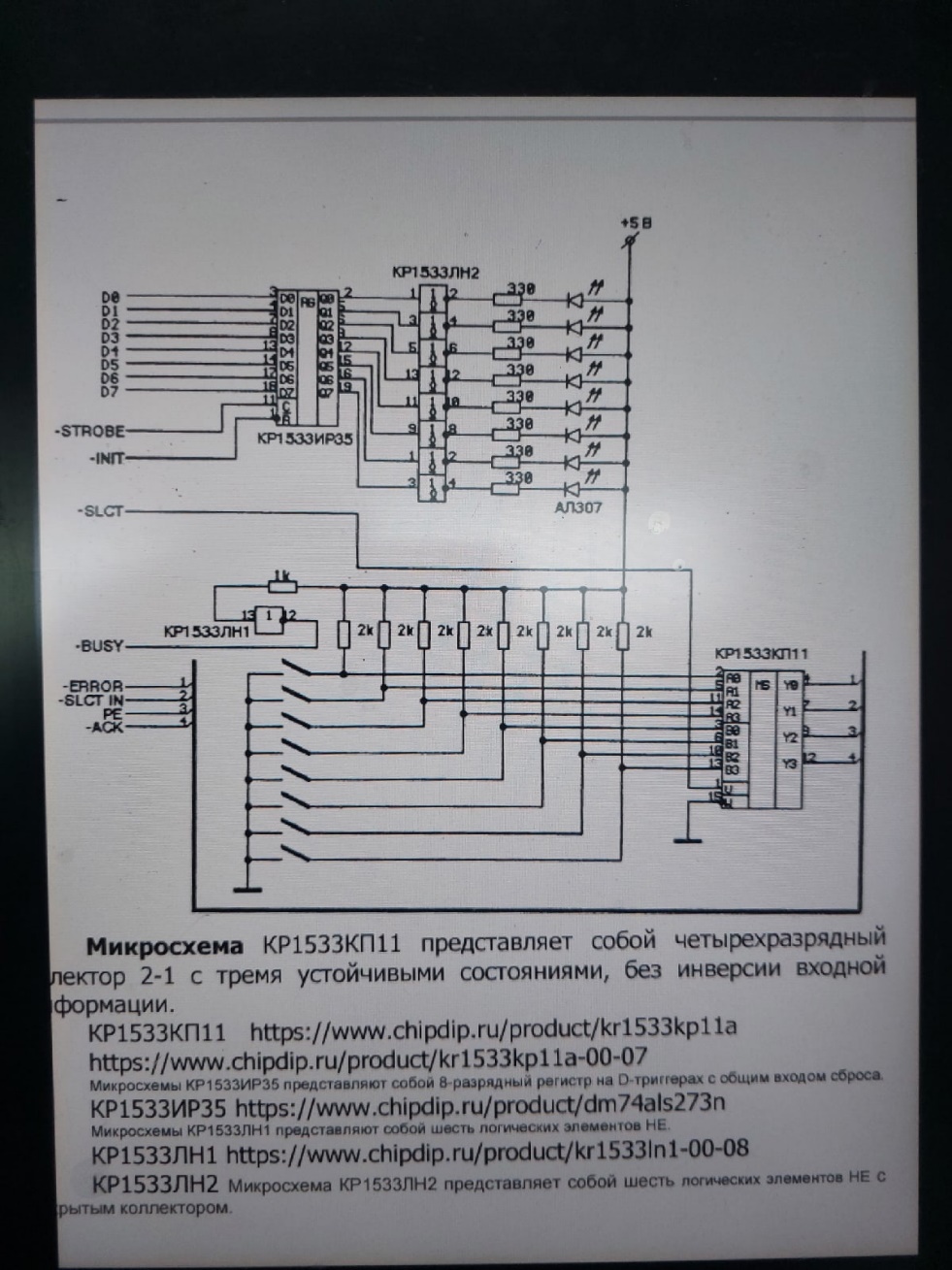
Москва2022

**Задание**

Написать программу для процессора i8086, которая будет управлять 4 цилиндрами согласно варианту по прерыванию IRQ7.

**Выполнение**

Можем управлять 3-мя регистрами: 037Аh – регистр управления, 0379h – регистр состояния, 0378h – регистра данных.



*Рис.1. Устройство контроллера Centronics*

На рис.2 представлено задание для Варианта 4.

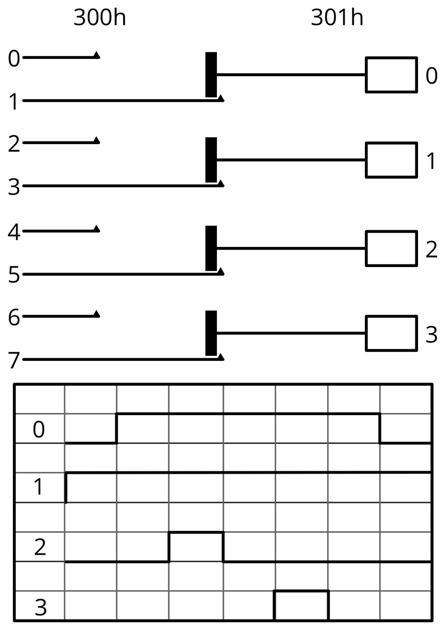
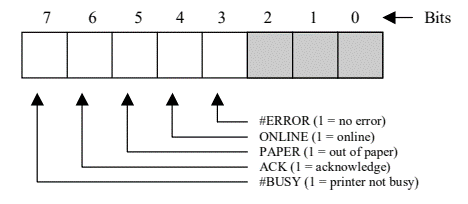
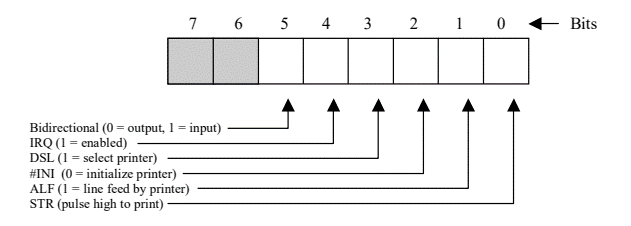


Рис. 2 Вариант 4

**

*Рис.3. Регистр состояния*

**

*Рис.4. Регистр управления*

В зависимости от состояния SLCT считываем из регистра состояния либо первые, либо вторые 4 бита - положения конечных выключателей. При помощи сдвига получаем 8-ми разрядное число, которое сравниваем с состояниями конечных выключателей из циклограммы. Как только получилась нужная комбинация, переходим к следующему такту.

**Листинг программы**

|  |  |  |  |
| --- | --- | --- | --- |
| **.model small**  **.stack 100h**  **.data**  **my\_mem DB 0**  **IRQ7 EQU 0Fh**  **offset\_interupt EQU 100h**  **.code**  **start:**  **mov ax, @data**  **mov ds, ax**  **cli**  **push es**  **xor ax, ax**  **mov es, ax**  **mov es:[IRQ7], offset\_interupt**  **mov es:[IRQ7+2], cs**  **pop es**  **sti**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010000b**  **out dx, ax**  **org offset\_interupt**  **mov dx,37AH**  **in ax, dx**  **and ax,00011000b**  **out dx,ax**  **mov dx,379H**  **in ax,dx**  **and ax,01111000b**  **mov bx,ax**  **shl bx,1**  **mov dx,37AH**  **in ax, dx**  **xor ax, 00001000b**  **out dx,ax**  **mov dx,379H**  **in ax,dx**  **and ax,01111000b**  **shr ax, 3**  **or ax,bx**  **test al, 01010101b**  **jz tact0**  **test al, 01011001b** | **jz tact1**  **test al, 01011010b**  **jz tact2**  **test al, 01101010b**  **jz farjmp1**  **test al, 10100101b**  **jz farjmp2**  **farjmp1:**  **jmp tact3**  **farjmp2:**  **jmp tact6**  **tact0:**  **mov dx, 378h**  **in ax, dx**  **or ax, 00000010b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **jmp exit**  **tact1:**  **mov al, my\_mem**  **and al, 00000010b**  **test al,11111101b**  **jz tact7**  **mov dx, 378h**  **in ax, dx**  **or ax, 00000011b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **or my\_mem,00000010b**  **jmp exit**  **tact7:**  **mov dx, 378h**  **in ax, dx**  **and ax, 00000000b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx** | **and ax, 00010001b**  **out dx, ax**  **and my\_mem, 00000000b**  **jmp exit**  **tact2:**  **mov al, my\_mem**  **and al, 00000100b**  **test al,11111011b**  **jz tact4**  **mov dx, 378h**  **in ax, dx**  **or ax, 00000111b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **or my\_mem,00000100b**  **jmp exit**  **tact3:**  **mov dx, 378h**  **in ax, dx**  **or ax, 00000011b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **jmp exit**  **tact4:**  **mov al,my\_mem**  **and al, 00100100b**  **test al,11011011b**  **jz tact5**  **mov dx, 378h**  **in ax, dx**  **and ax, 00001011b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **or my\_mem,00100100b** | **jmp exit**  **tact5:**  **mov dx, 378h**  **in ax, dx**  **and ax, 00000011b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **jmp exit**  **tact6:**  **mov dx, 378h**  **in ax, dx**  **and ax, 00000010b**  **out dx, ax**  **mov dx, 37Ah**  **in ax, dx**  **and ax, 00010001b**  **out dx, ax**  **jmp exit**  **exit:**  **mov al, 20h**  **out 20h, al**  **iret**  **end** |