

Лекция №4
Программирование
циклических алгоритмов на
ассемблере

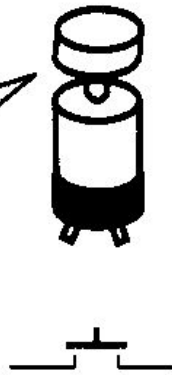
Условие задачи

После замыкания кнопки поочередно парами загораются 16 симметрично расположенных светодиодов (по 8 с каждой стороны). После того как все светодиоды загорятся, они должны одновременно погаснуть. При повторном нажатии на кнопку все повторяется. Если кнопка не нажата – светодиоды не горят.

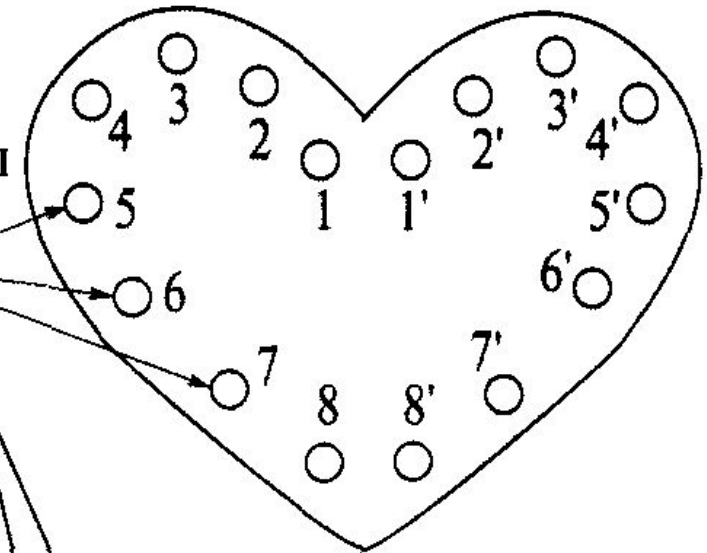
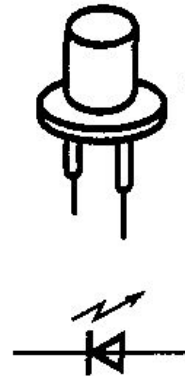
Описание устройства

Выключатель
с нормально
разомкнутыми
контактами

При нажатии
кнопки
выключателя
контакты
замыкаются, в
свободном
состоянии —
размыкаются

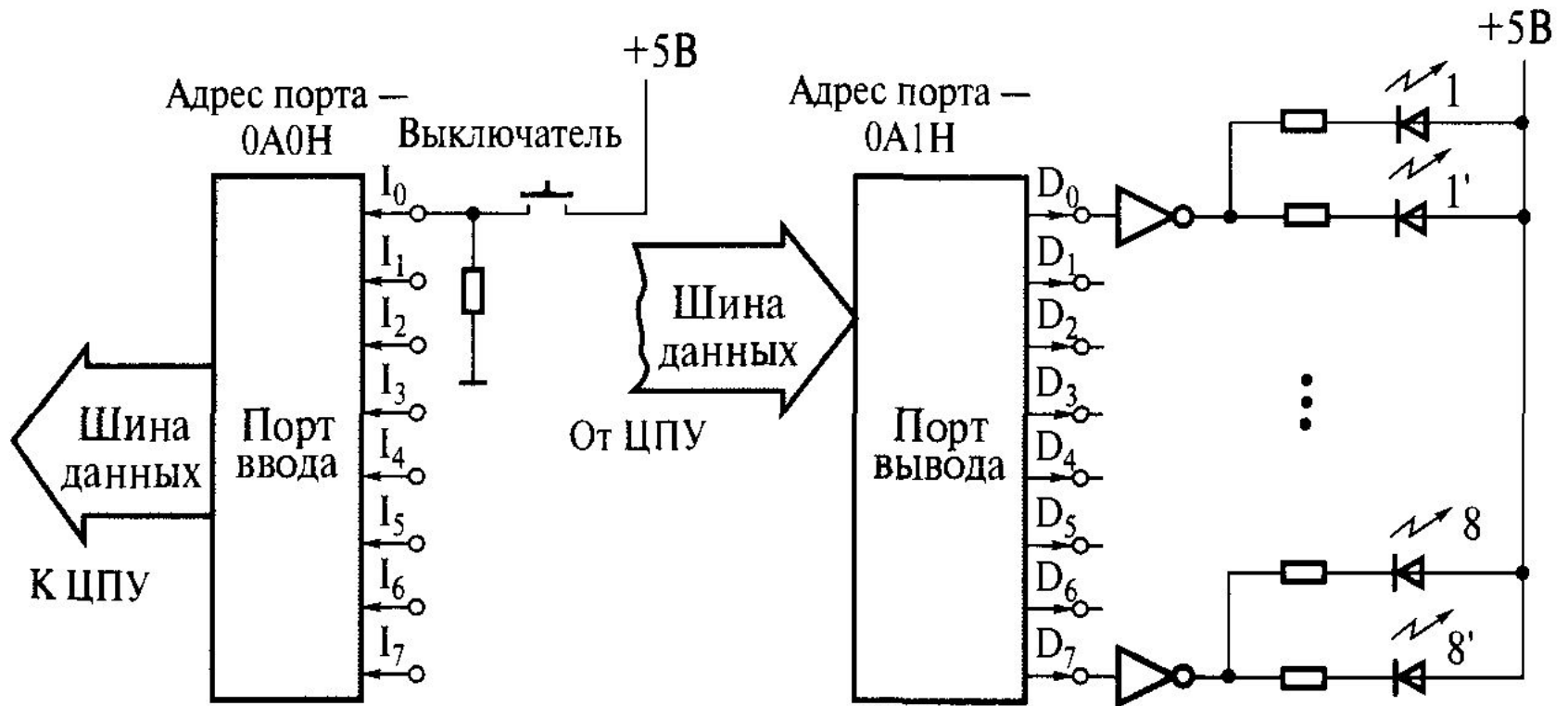


Светодиоды



Загорается при
протекании тока

Схема устройства



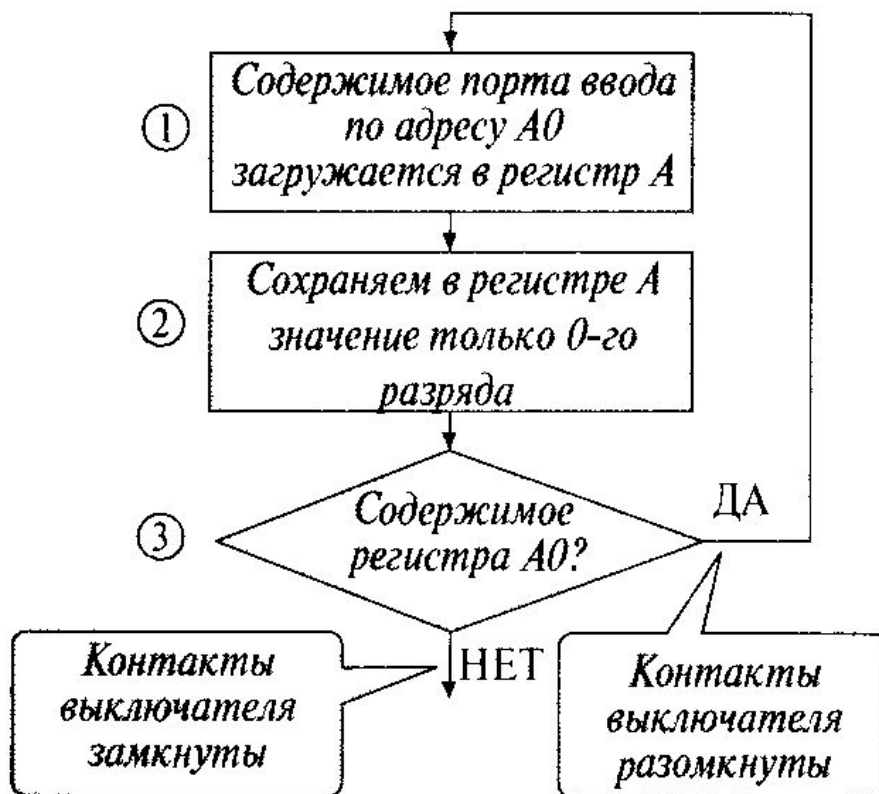
Описание алгоритма программы

Программа состоит из трех частей:

- 1. Управление кнопкой.**
- 2. Управление светодиодами.**
- 3. Введение задержки.**

Управление кнопкой

Схема алгоритма



Программа управления кнопкой

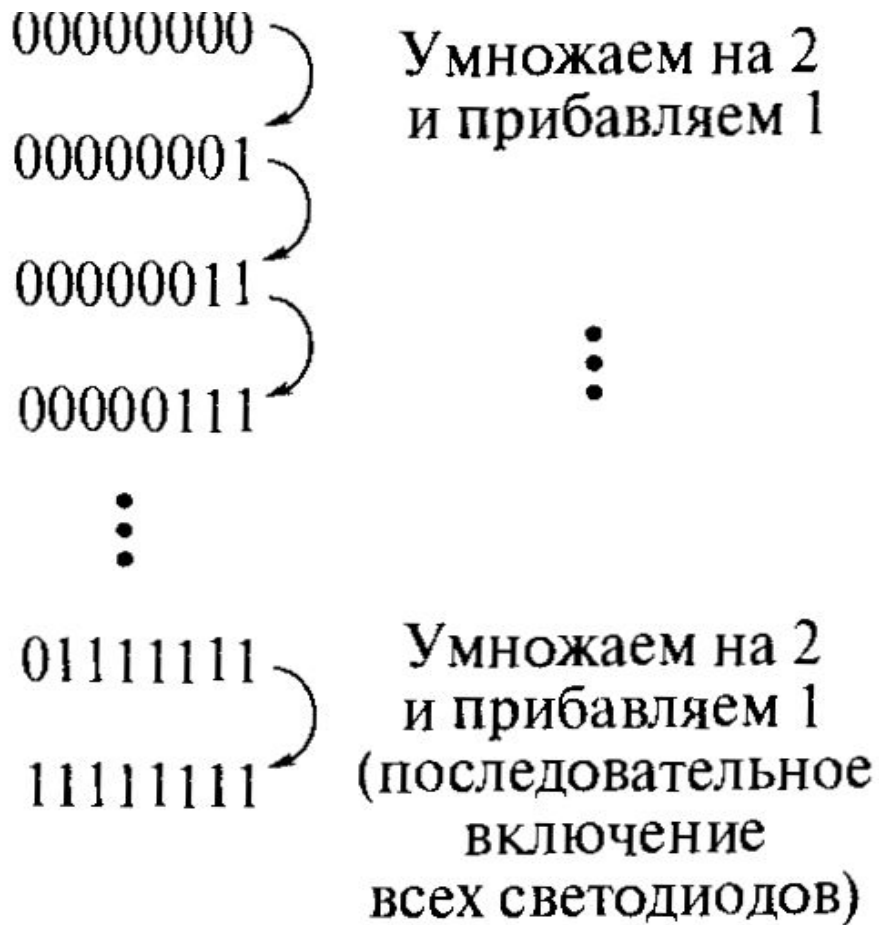
Программа в мнемосодах

```
LOOP1: IN  A, (0A0H); ①  
        AND 01H ; ②  
        JP  Z, LOOP1 ; ③
```

**~ Задача, выполняемая
данной программой ~**

Проверяем состояние выключателя,
ожидаем момента замыкания контактов
(фиксируем состояние «включено»)

Управление светодиодами



Программа управления светодиодами (1)

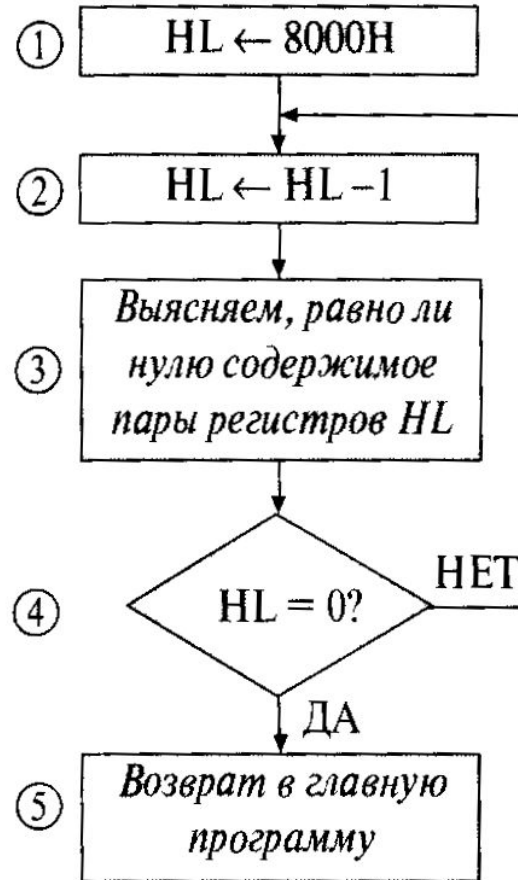
```
XOR A          ; «Обнуляем»  
                регистр A  
-----  
LOOP2 : ADD A, A      ; Содержимое  
                    регистра A  
                    умножаем на 2  
        INC A        ; К содержимому  
                    регистра A  
                    прибавляем 1  
        OUT (0A1H), A ; Включаем диод  
  
        CP   OFFH    ; } Повторяем цикл  
                    ; } до тех пор, пока  
                    ; } не включатся  
        JP   NZ, LOOP2 ; } все светодиоды
```

Цикл

Программа управления светодиодами (2)

```
{ XOR A          ; «Обнуляем»  
  ;             регистр A  
  OUT (0A1H) A ; Выключаем  
                светодиоды
```

Введение задержки



Подпрограмма задержки

DELAY : LD HL, 8000H ①

LOOP3 : DEC HL ②

LD A, H } ③
OR L }

JP NZ, LOOP3 ④

RET ⑤

Объединенная программа (1)

```
LD SP, 1000H  
LOOP1: IN A, (0A0H)  
AND 01H  
JP Z, LOOP1  
XOR A
```

①

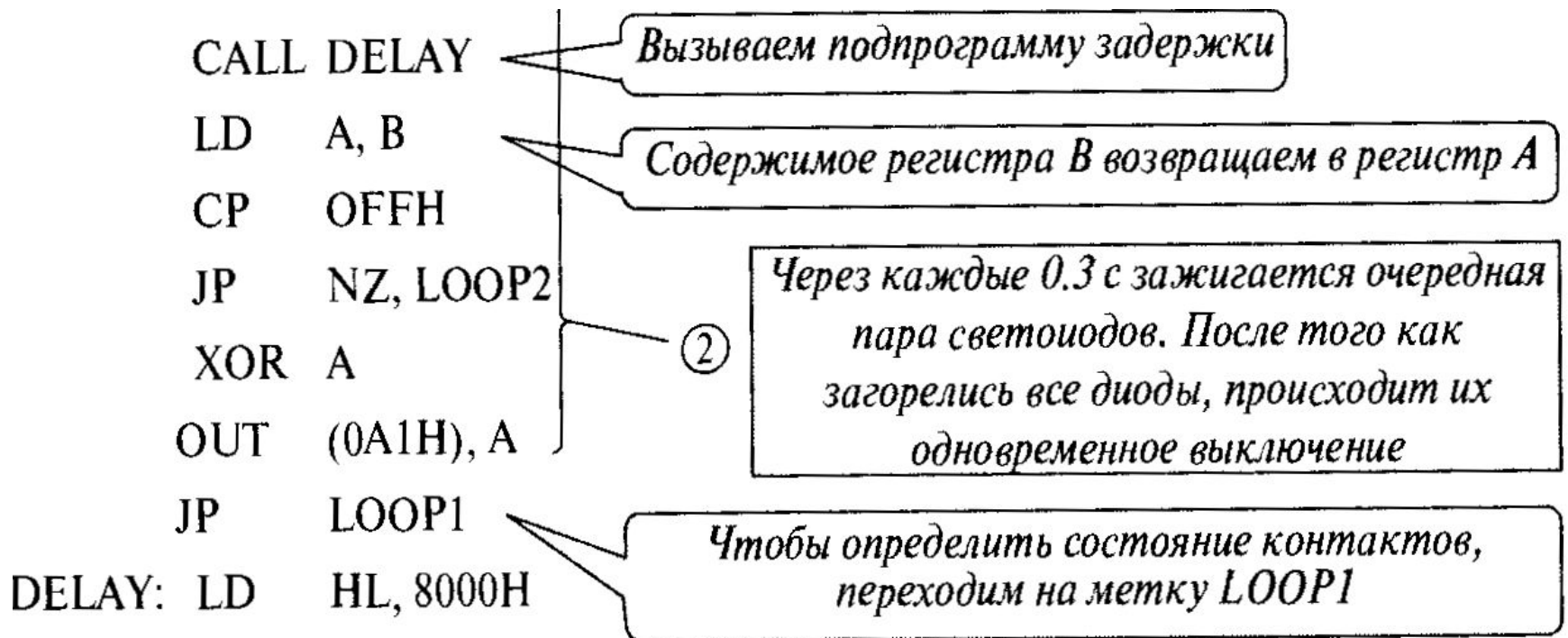
*Ждем замыкания
контактов переключателя*

Объединенная программа (2)

```
LOOP2: ADD  A, A  
        INC  A  
        OUT (0A1H), A  
        LD   B, A
```

Содержимое регистра A «прячем» в регистре B

Объединенная программа (3)



Объединенная программа (4)

```
DELAY: LD    HL, 8000H  
LOOP3: DEC  HL  
      LD   A, H  
      OR  L  
      JP  NZ, LOOP3  
      RET
```

③

Считаем до 32768, чтобы
«убить» время

Машинный код (1)

| <i>Начальные адреса программы</i> | | <i>Программа в мнемокодах</i> | |
|-----------------------------------|-----------------------------------|-------------------------------|-------------|
| <i>Адреса</i> | <i>Программа в машинных кодах</i> | | |
| | | ORG 0000H | } Директивы |
| | | PORT1 EQU 0A0H | |
| | | PORT2 EQU 0A1H | |
| 0000 | 31 00 10 | LD SP, 1000H | |
| 0003 | DB A0 | LOOP1: IN A, (PORT1) | |
| 0005 | E6 01 | AND 01H | |
| 0007 | CA 03 00 | JP Z, LOOP1 | |
| 000A | AF | XOR A | |

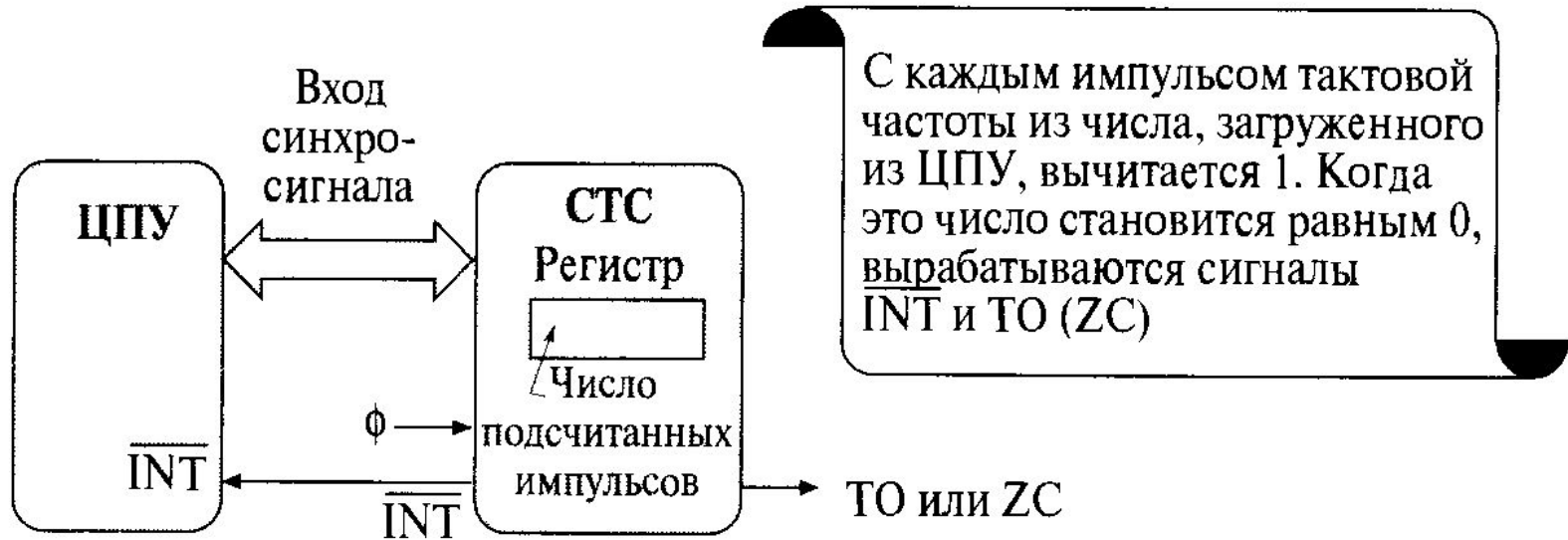
Машинный код (2)

| | | |
|------|----------|-----------------|
| 000B | 87 | LOOP2: ADD A, A |
| 000C | 3C | INC A |
| 000D | D3 A1 | OUT (PORT2), A |
| 000F | 47 | LD B, A |
| 0010 | CD 1F 00 | CALL DELAY |
| 0013 | 78 | LD A, B |
| 0014 | FE FF | CP OFFH |
| 0016 | C2 0B 00 | JP NZ, LOOP2 |
| 0019 | AF | XOR A |
| 001A | D3 A1 | OUT (PORT2), A |
| 001C | C3 03 00 | JP LOOP1 |

Машинный код (3)

| | | | |
|-------|----|-------|---------------------|
| 0019 | AF | | XOR A |
| 001A | D3 | A1 | OUT (PORT2), A |
| 001C | C3 | 03 00 | JP LOOP1 |
| 001F | 21 | 00 80 | DELAY: LD HL, 8000H |
| 0022 | 2B | | LOOP3: DEC HL |
| 0023 | 7C | | LD A, L |
| 0024 | B5 | | OR L |
| 0025 | C2 | 22 00 | JP NZ, LOOP3 |
| 0028 | C9 | | RET |
| <hr/> | | | |
| | | | END Директива |

Счетчик-таймер



Таймер

.....

Осуществляет задержку, определяемую числом прошедших импульсов

Счетчик

.....

Подсчитывает импульсы, поданные на вход (это число может быть считано ЦПУ)

