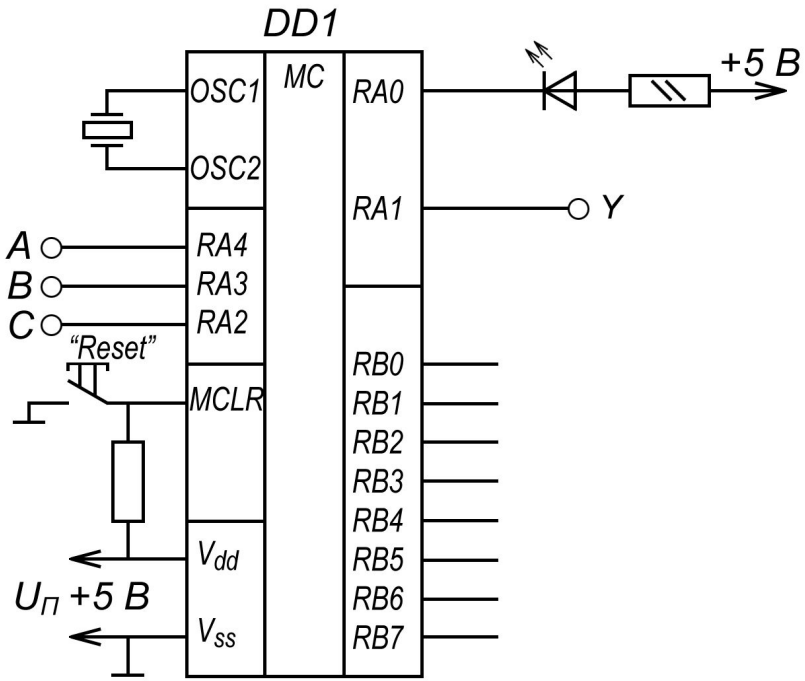
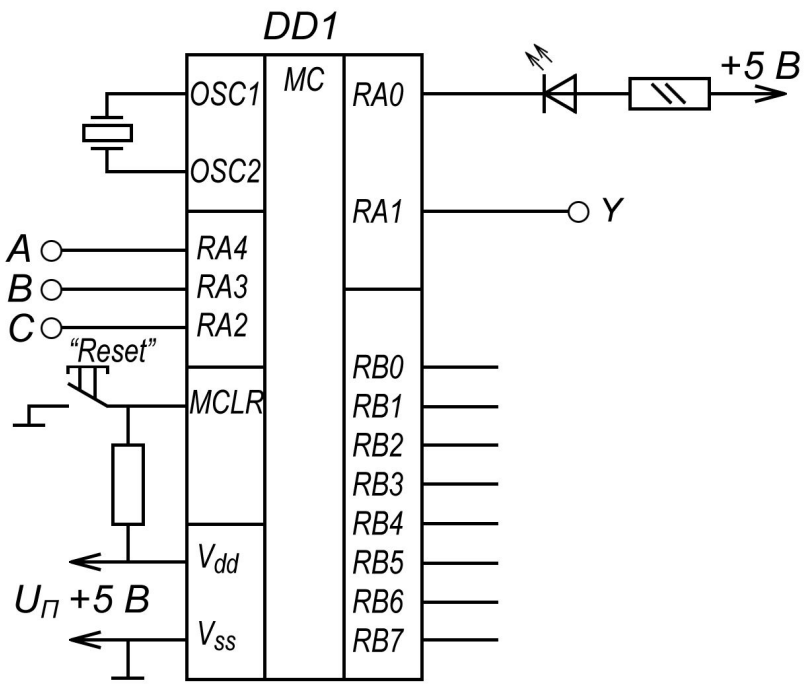
# Электроника и микропроцессоры. Модуль 2, Тема №5
# «Построение микропроцессорной системы на основе микроконтроллера»
## Разработка программы

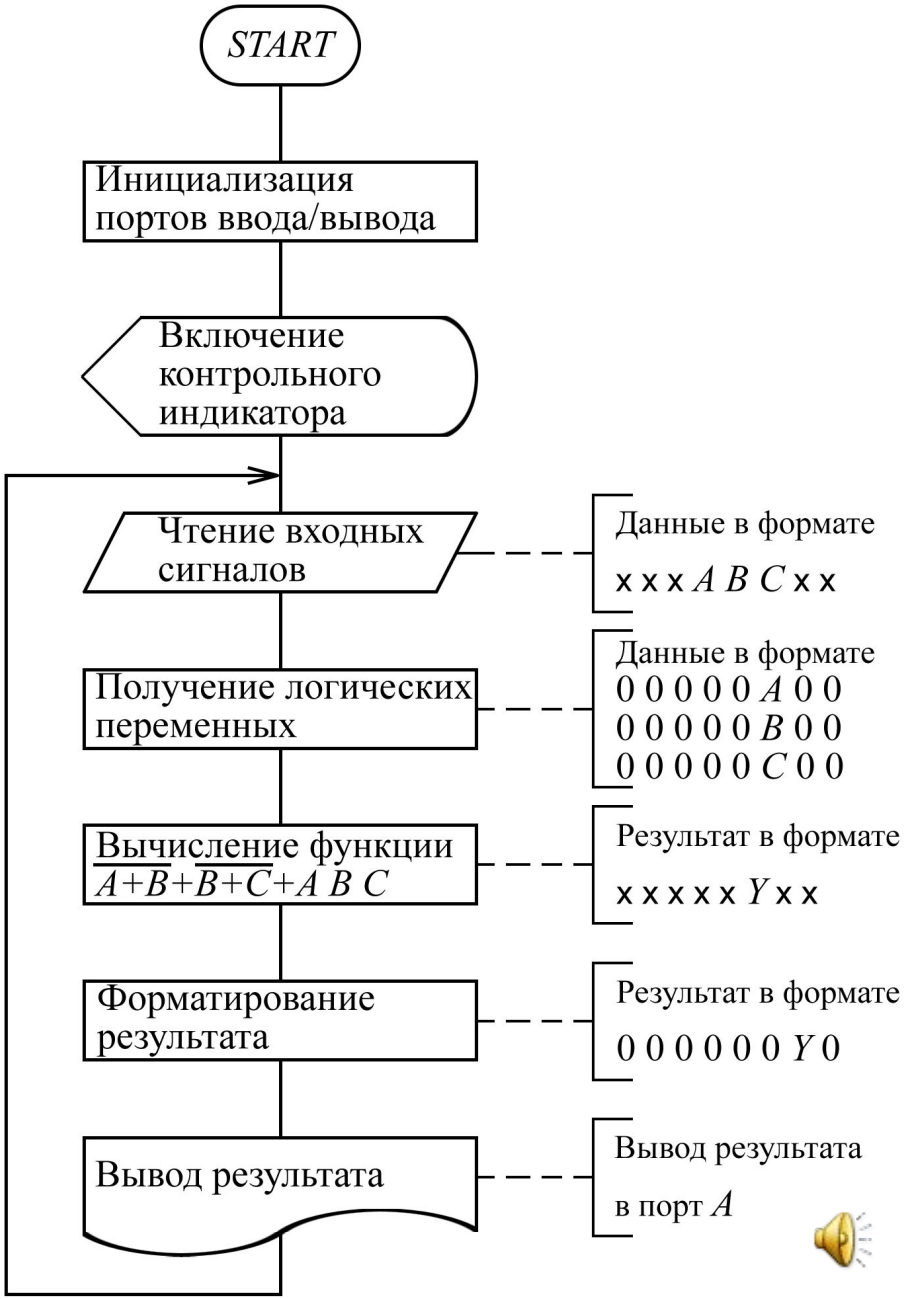
*То что не удается запаять приходится программировать.*

(С) народная мудрость.
DI Halt - AVR Учебный курс



Электроника и микропроцессоры. Модуль 2, Тема №5
«Построение микропроцессорной системы на основе микроконтроллера»
Разработка программы




```
«Прошивка» FLASH ROM в формате Intel HEX.
:020000040000FA
:10000000031383161C308500FF308600831205140D
:1000100005089000043 98E00100808398D008D0CF9
:10002000100810398C008C0C8C0C0C080D048F00FF
:100030008F090D080E04043A8F040C080D050E05F7
:100040008F040F0C023985000828663092 00F930C1
:1000500091000000000000910B2B280000920B83
:040060002728080045
:02400E00F13F80
:00000001FF
```

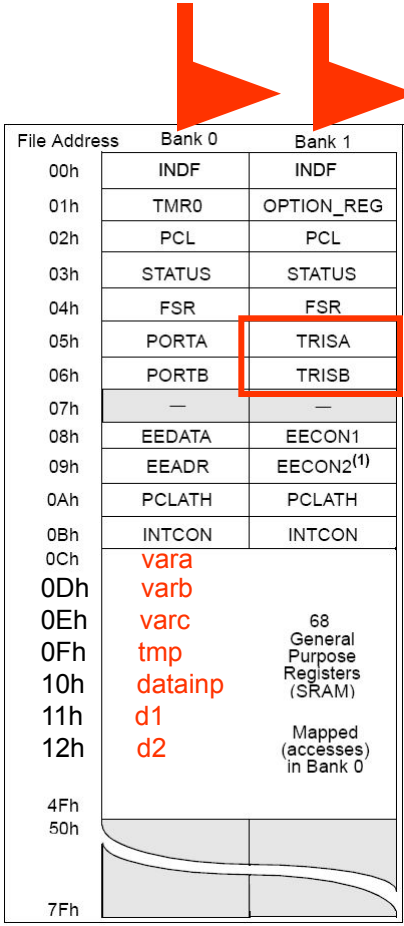
# Электроника и микропроцессоры. Модуль 2, Тема №5
# «Построение микропроцессорной системы на основе микроконтроллера»
# Разработка программы

```
;*****************************************************************
;    Filename:      example_dz-pic.asm
; Программа вводит три сигнала через порт A (A-RA4, B-RA3, C-RA2),
; вычисляет функцию трех логических переменных Y
; и выводит результат в порт A (Y-RA1) в прямом виде.
;*****************************************************************
;
list p=16F84A    ; list directive to define processor
#include <p16F84a.inc> ; processor specific variable definitions
__CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC
;----- назначаем имена переменных и соответствующие им адреса
vara  EQU 0x0C   ;
varb  EQU 0x0D   ; - переменные
varc  EQU 0x0E   ;
tmp   EQU 0x0F   ; переменная для промежуточных результатов
datainp    EQU 0x10   ; переменная, содержащая текущее состояние сигналов
d1    EQU 0x11   ; переменная для подпрограммы - задержки
d2    EQU 0x12   ; переменная для подпрограммы - задержки
;-----------------------------------------------------------------
ORG 0x000        ; processor reset vector
bcf STATUS,RP1
bsf STATUS,RP0   = bcf 0x03,
movlw 0x1C       0x06
movwf TRISA      ; RA4, RA3, RA2 - на ввод, RA1, RA0 - на вывод
movlw 0xFF       ; управляющее слово для порта B
movwf TRISB      ; все на ввод
bcf STATUS,RP0   ; возвращаемся в банк 0
;-----------------------------------------------------------------
; Приветливо мигнем индикатором на PORTA[0] и оставим его включенным -
; - не обязательный, сервисный фрагмент программы
bcf   PORTA, 0   ; зажгли
call  del_200    ; подождали 200 мс
bsf   PORTA, 0   ; погасили
call  del_200    ; подождали 200 мс
bcf   PORTA, 0   ; зажгли
```

| 0 1 0 0 | b b b | f f f f f f f |
|---|---|---|
| bcf | RP1 | STATUS |

| 0 1 | 0 0 | 1 1 | 0 | 0 0 0 | 0 0 1 1 |
|---|---|---|---|---|---|

=0x1303



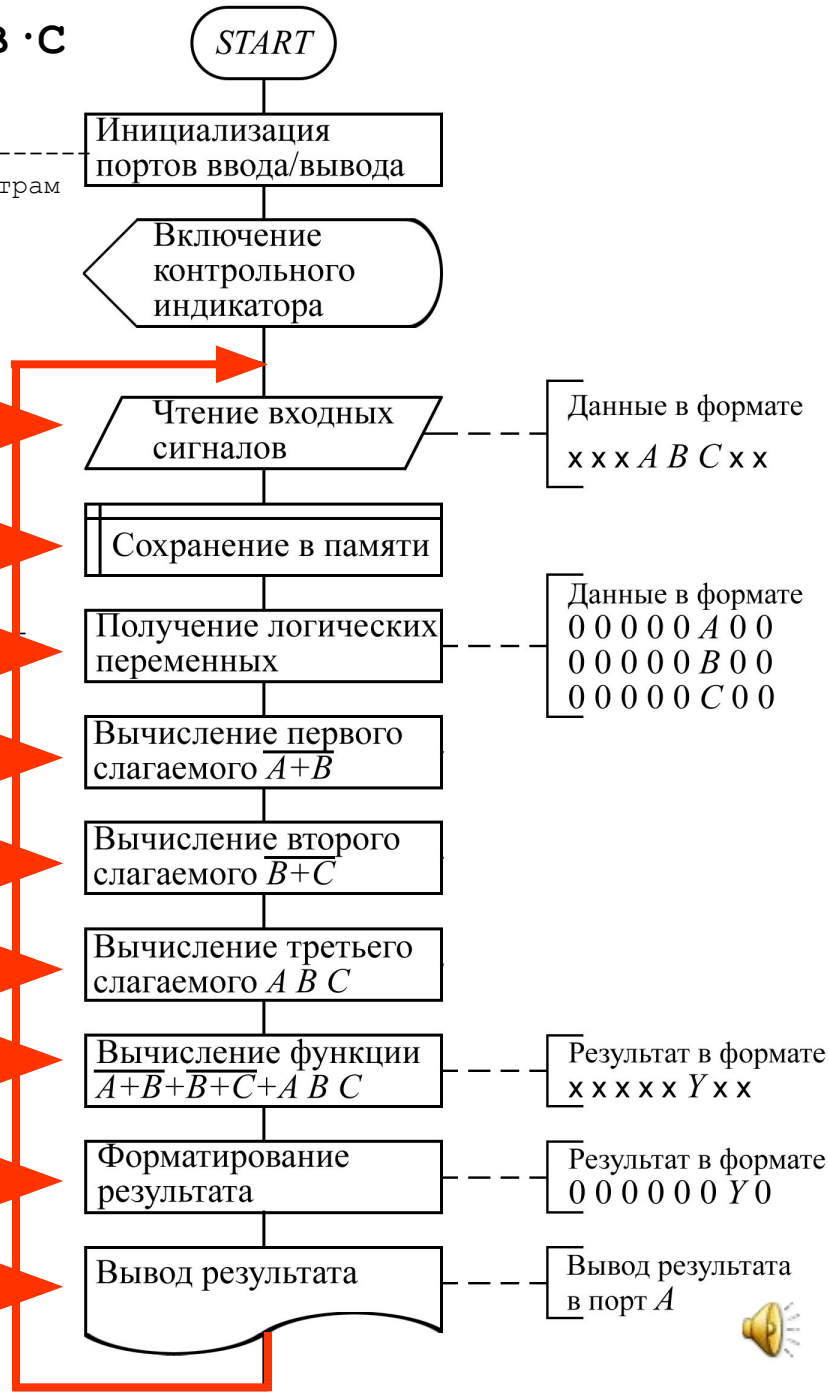

| STATUS | IRP | | 0 | | 1 | | -TO | -PD | Z | DC | C |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Бит 7 | | | | | | | | | | Бит 0 |

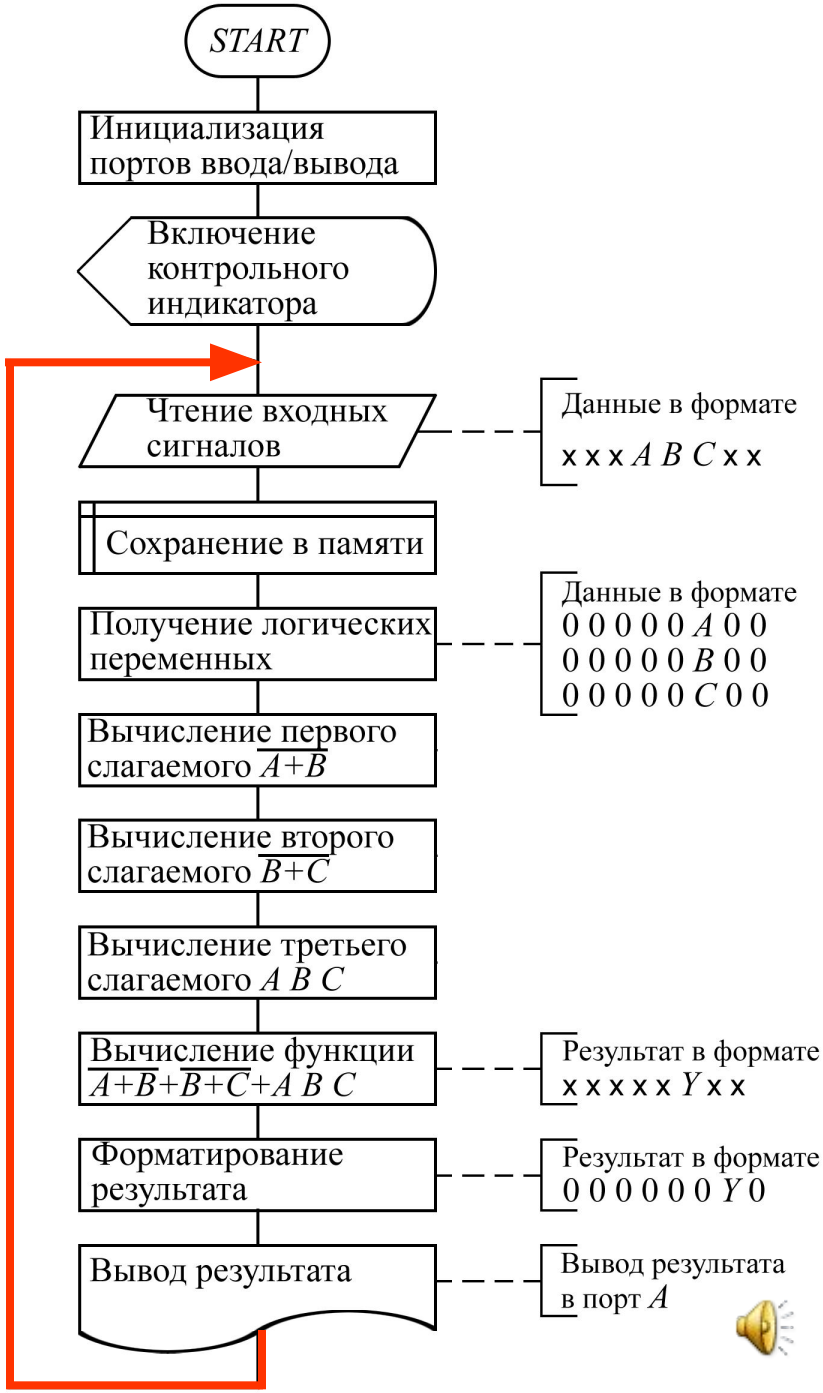
$$Y = \overline{A+B} + \overline{B+C} + A \cdot B \cdot C$$

```
; ---------------  Основной исполнительный цикл  ----------------------
; получаем комбинацию переменных и раскладываем их по разным регистрам
start  movf PORTA, w    ; читаем данные из порта A
movwf datainp           ; сохраняем
andlw 0x04  ; накладываем  маску 00000100 - выделяем C
movwf varc  ; сохраняем в varc
movf  datainp, w  ; восстанавливаем исходные данные в аккумулятор
andlw 0x08  ; накладываем  маску 00001000 - выделяем B
movwf varb  ; сохраняем в varb
rrf   varb, f     ; сдвиг B вправо - выравниваем с C
movf  datainp, w  ; восстанавливаем исходные данные в аккумулятор
andlw 0x10  ; накладываем  маску 00010000 - выделяем A
movwf vara  ; сохраняем в vara
rrf   vara, f     ; сдвиг A вправо
rrf   vara, f     ; сдвиг A вправо - выравниваем с C
; ---------------- вычисляем функцию ----------------------------
movf  vara, w     ; A - в аккумулятор
iorwf varb, w     ; в аккумуляторе A+B
movwf tmp         ; сохраняем в tmp
comf  tmp, f      ; инвертируем A+B с сохранением в tmp
movf  varb, w     ; B - в аккумулятор
iorwf varc, w     ; в аккумуляторе B+C
xorlw 0x04  ; инвертируем B+C с сохранением в w
iorwf tmp, f  ; !(A+B)+!(B+C) с сохранением в tmp
movf  vara, w     ; A - в аккумулятор
andwf varb, w     ; конъюнкция AB в аккумуляторе
andwf varc, w     ; конъюнкция ABC в аккумуляторе
iorwf tmp, f  ; !(A+B)+!(B+C)+ABC с сохранением в tmp - РЕЗУЛЬТАТ
rrf   tmp, w  ; сдвиг вправо - выравниваем с Y с сохранением в w
andlw 0x02  ; накладываем  маску 00000010 - чистим результат
            ; перед выводом в порт
movwf PORTA ; выводим результат в порт
goto  start ; Зацикливаем программу
```

метка команда параметры

| & | X | X | A | B | C | X | X | datainp |
|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | B | 0 | 0 | - varb |
| | 0 | 0 | 0 | 0 | 0 | C | 0 | - varc |
| | 0 | 0 | 0 | 0 | 0 | Y | 0 | |



```
; ------ Итого 30 машинных циклов, т.е. при тактовой
частоте 4 МГц время выполнения цикла 30 мкс.
; --- Подпрограмма Задержка 200 мс ---
del_200
movlw 0xCC
movwf d2
d200_2 movlw 0xF9
movwf d1
nop
nop
d200_1 nop
decfsz d1,f
goto d200_1
nop
decfsz d2,f
goto d200_2
return
; --- end of DEL_200 ---
END
```

```
«Прошивка» FLASH ROM в формате Intel HEX.
:020000040000FA
:10000000031383161C308500FF308600831205140D
:1000100005089000043 98E00100808398D008D0CF9
:10002000100810398C008C0C8C0C0C080D048F00FF
:100030008F090D080E04043A8F040C080D050E05F7
:100040008F040F0C023985000828663092 00F930C1
:10005000910000000000000910B2B280000920B83
:040060002728080045
:02400E00F13F80
:00000001FF
```