

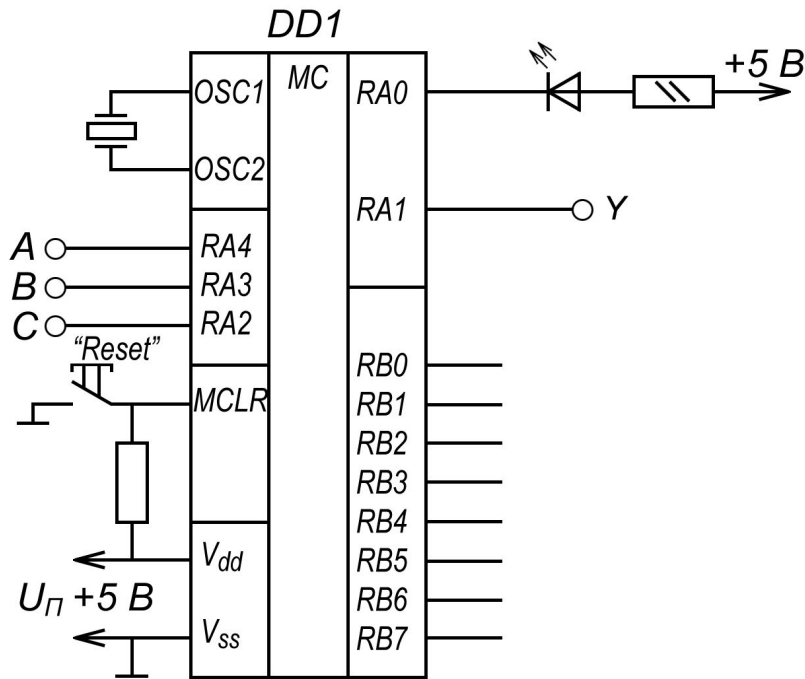
Электроника и микропроцессоры. Модуль 2, Тема №5

«Построение микропроцессорной системы на основе микроконтроллера»

Разработка программы

То что не удается запаять приходится программировать.

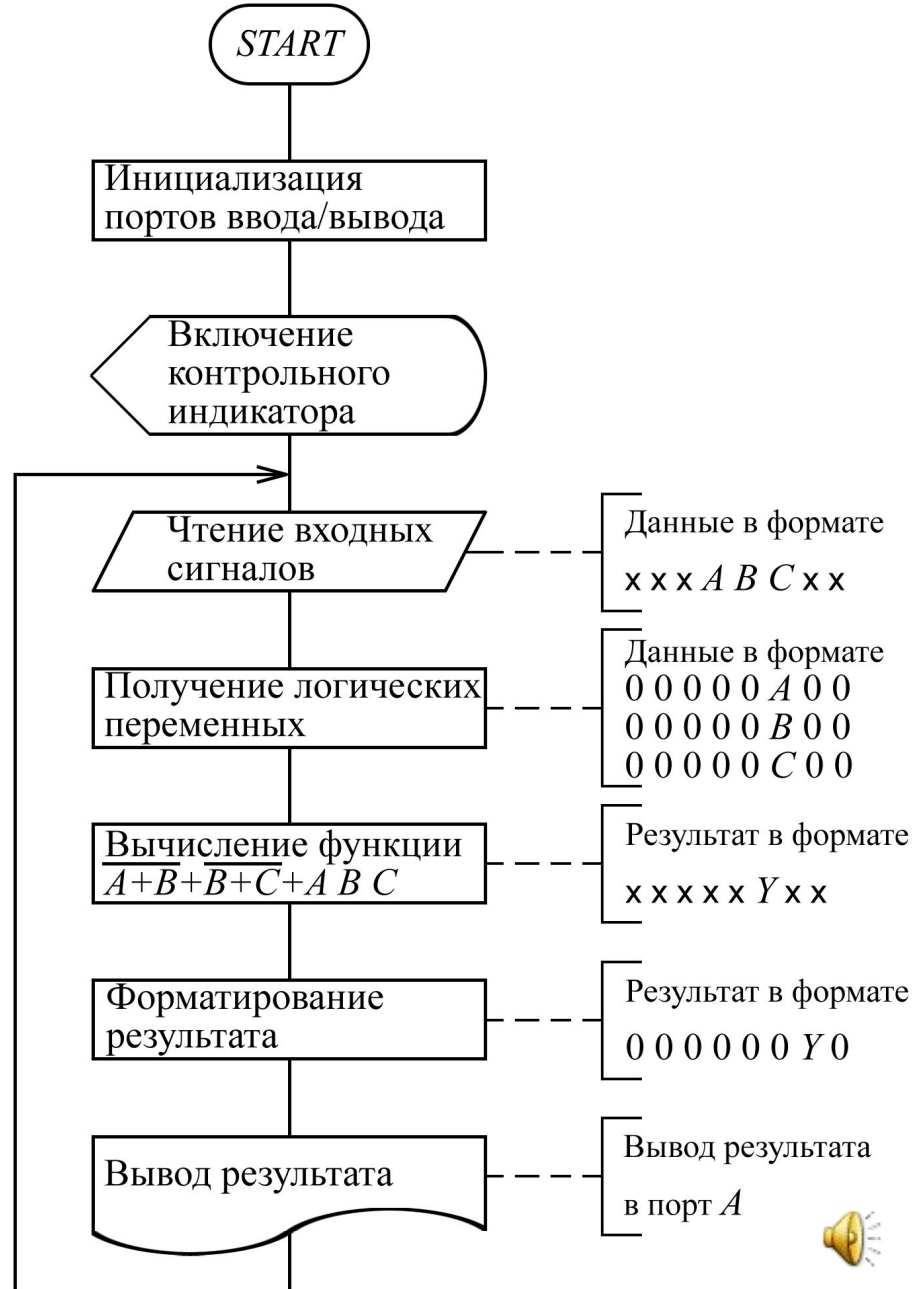
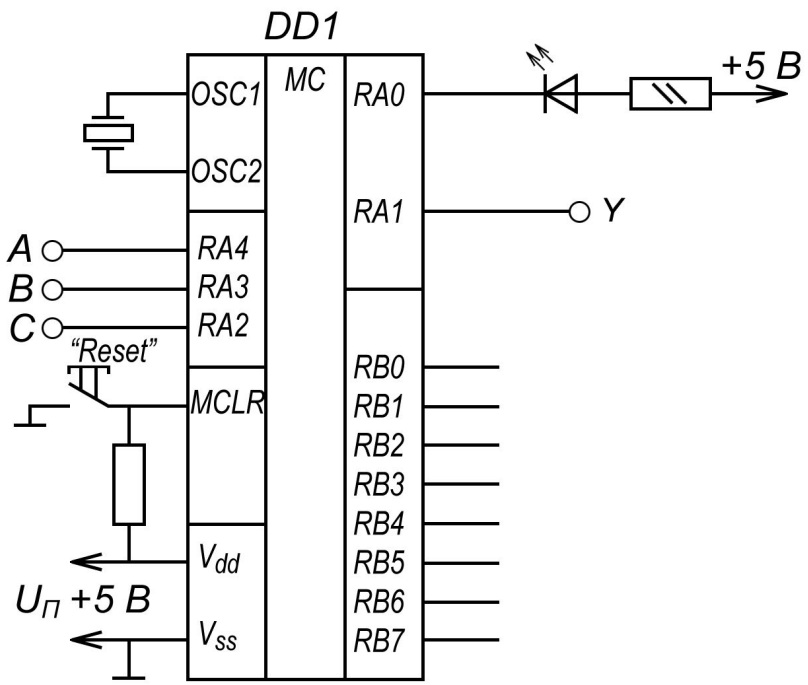
(С) народная мудрость.
DI Halt - AVR Учебный курс



Электроника и микропроцессоры. Модуль 2, Тема №5

«Построение микропроцессорной системы на основе микроконтроллера»

Разработка программы



```

«Прошивка» FLASH ROM в формате Intel HEX.
:020000040000FA
:10000000031383161C308500FF308600831205140D
:100010000508900004398E00100808398D008D0CF9
:10002000100810398C008C0C8C0C0C080D048F00FF
:100030008F090D080E04043A8F040C080D050E05F7
:100040008F040F0C02398500082866309200F930C1
:1000500091000000000000000910B2B280000920B83
:040060002728080045
:02400E00F13F80
:00000001FF
  
```



Электроника и микропроцессоры. Модуль 2, Тема №5

«Построение микропроцессорной системы на основе микроконтроллера»

Разработка программы

```

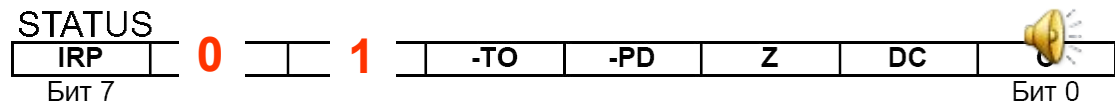
;*****
;   Filename:      example_dz-pic.asm
; Программа вводит три сигнала через порт А (А-RA4, В-RA3, С-RA2),
; вычисляет функцию трех логических переменных Y
; и выводит результат в порт А (Y-RA1) в прямом виде.
;*****
;
list p=16F84A    ; list directive to define processor
#include <p16F84a.inc> ; 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
movlw 0x1C
movwf TRISA
movlw 0xFF
movwf TRISB
bcf STATUS,RP0
;
; 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

```



=0x1303

File Address	Bank 0	Bank 1
00h	INDF	INDF
01h	TMR0	OPTION_REG
02h	PCL	PCL
03h	STATUS	STATUS
04h	FSR	FSR
05h	PORTA	TRISA
06h	PORTB	TRISB
07h	—	—
08h	EEDATA	EECON1
09h	EEADR	EECON2 ⁽¹⁾
0Ah	PCLATH	PCLATH
0Bh	INTCON	INTCON
0Ch	vara	68 General Purpose Registers (SRAM)
0Dh	varb	
0Eh	varc	
0Fh	tmp	
10h	datainp	
11h	d1	Mapped (accesses) in Bank 0
12h	d2	
4Fh	[Shaded area]	
50h		
7Fh	[Shaded area]	



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

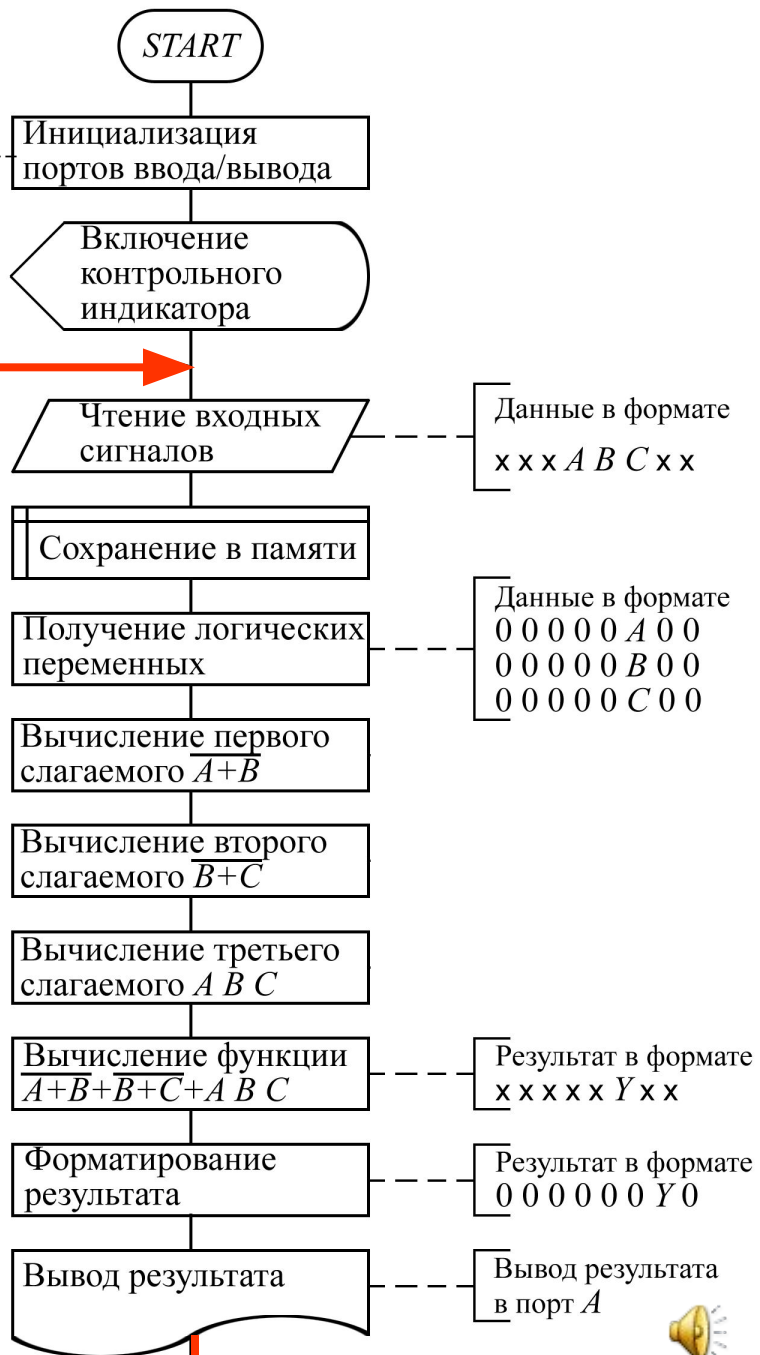
```

; ----- Основной исполнительный цикл -----
; получаем комбинацию переменных и раскладываем их по разным регистрам
start movf PORTA, w ; читаем данные из порта A
movwf datainp ; сохраняем
anlwf 0x08 ; накладываем маску 00000100 - выделяем C
movf varc, w ; сохраняем в varc
movf datainp, w ; восстанавливаем исходные данные в аккумулятор
anlwf 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, w ; 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 varc с сохранением в w
andlw 0x02 ; накладываем маску 00000010 - чистим результат
movwf PORTA ; выводим результат в порт
goto start ; Зацикливаем программу

```

↑ метка
↑ команда
↑ параметры

&	X	X	X	A	B	C	X	X	datainp
0	0	0	0	0	0	0	0	0	
0	0	0	0	A	B	0	0	0	
0	0	0	0	A	B	0	0	0	- vara
0	0	0	0	0	B	0	0	0	- varb
0	0	0	0	0	C	0	0	0	- varc
0	0	0	0	0	0	0	0	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
:100010000508900004398E00100808398D008D0CF9
:10002000100810398C008C0C8C0C0C080D048F00FF
:100030008F090D080E04043A8F040C080D050E05F7
:100040008F040F0C02398500082866309200F930C1
:100050009100000000000000910B2B280000920B83
:040060002728080045
:02400E00F13F80
:00000001FF

```

