

Домашнее задание N1 по курсу "Электроника и МПТ".

Составить принципиальную схему цифрового устройства, используя любые реальные элементы ТТЛ – логики, таблица истинности которого соответствует заданной. Минимизацию функции произвести двумя способами (по правилам алгебры логики и методом карт Карно). Оформить задание с соблюдением ГОСТ 2.743-91.

Пример решения задачи 1-го ДЗ

Задана в виде таблицы истинности логическая функция Y трех переменных A, B, C .

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Функция принимает единичное значение при следующих произведениях переменных:

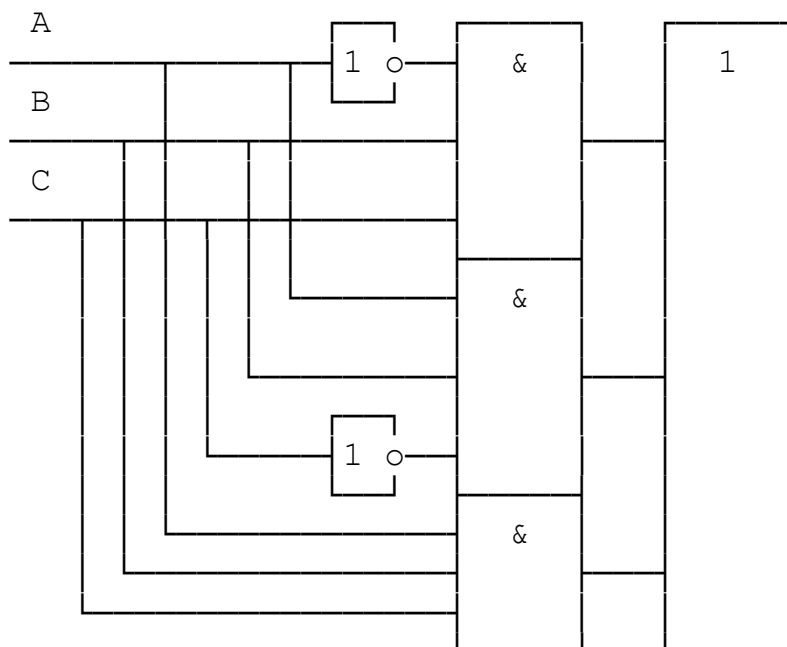
$$\bar{A}BC, A\bar{B}\bar{C}, ABC$$

Каждое из произведений переменных для которых значение функции истинно, носит название МИНТЕРМА.

В соответствии с этим, функция может быть записана в виде уравнения:

$$Y = \bar{A}BC + A\bar{B}\bar{C} + ABC$$

Если при такой записи каждое слагаемое содержит произведения всех переменных или их отрицаний, то такую форму представления функции называют СОВЕРШЕННОЙ ДИЗЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМОЙ или ПЕРВОЙ СТАНДАРТНОЙ ФОРМОЙ. Точно так же можно выделять ложные (нулевые) значения функции. Если функция дана в виде произведения (конъюнкции) сумм переменных или их отрицаний, то такую форму представления функции называют СОВЕРШЕННОЙ КОНЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМОЙ или ВТОРОЙ СТАНДАРТНОЙ ФОРМОЙ.



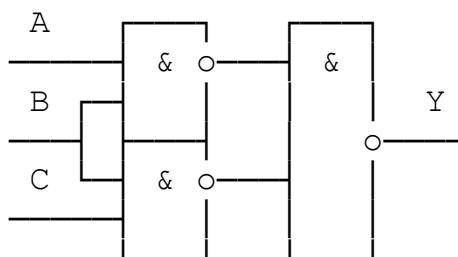
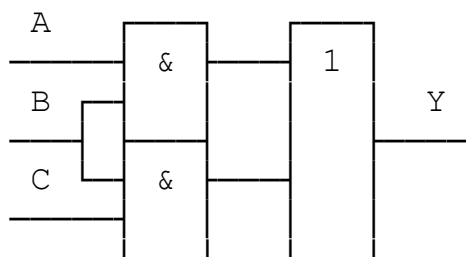
Для схемной реализации полученной функции потребуется:
 У 3 схемы, выполняющие функцию ЗИ,
 1 схема ЗИЛИ и
 2 схемы НЕ.

Пользуясь правилами алгебры логики упростим полученную функцию.

$$Y = \bar{A}BC + A\bar{B}C + ABC = \bar{A}BC + AB(\bar{C} + C) = \bar{A}BC + AB = B(\bar{A}C + A) =$$

$$= (\text{правило 15}) = B(\bar{A} + A)(C + A) = B(C + A) = BC + AB$$

Полученная функция и ее схемная реализация значительно проще исходных.



$$Y = \overline{\overline{AB + BC}} = \overline{\overline{AB} * \overline{BC}} - \text{для построения схемы на одинаковых элементах}$$

Можно было представить ту же самую функцию в СОВЕРШЕННОЙ КОНЪЮНКТИВНОЙ НОРМАЛЬНОЙ ФОРМЕ в виде произведения сумм:

$$Y = (A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + C)(\bar{A} + B + \bar{C})$$

Такая запись слишком сложна для минимизации и реализации.

Разработан метод минимизации логических функций, как бы автоматизирующий процедуру поиска "склеивающихся слагаемых" - **метод КАРТ КАРНО**.

Карта Карно - это таблица имеющая ячейки для всех возможных минтермов функции. Можно построить карты Карно для функций,

минтермы которых содержат два, три и более переменных (обычно не более 5...6). Составим карту Карно для функции 2-х переменных: Вдоль верхней грани проставлены возможные значения

	A	\bar{A}	переменной A, вдоль левой боковой грани -
B	AB	$\bar{A}B$	возможные значения переменной B. В каждой
\bar{B}	$A\bar{B}$	$\bar{A}\bar{B}$	клетке изображают один из возможных мин-

термов: AB, $\bar{A}B$, $A\bar{B}$, $\bar{A}\bar{B}$. Если какой-то из этих минтермов в совершенной дизъюнктивной нормальной форме записи функции присутствует, то в соответствующей клетке карты Карно ставится "1", если нет - то "0".

Составим карты Карно для функций 3-х и 4-х переменных:

	$\bar{\bar{A}}\bar{B}$	$\bar{A}\bar{B}$	$A\bar{B}$	$\bar{A}B$
\bar{C}	$\bar{\bar{A}}\bar{B}\bar{C}$	$\bar{A}\bar{B}\bar{C}$	$A\bar{B}\bar{C}$	$\bar{A}B\bar{C}$
C	$\bar{\bar{A}}\bar{B}C$	$\bar{A}\bar{B}C$	$A\bar{B}C$	$\bar{A}BC$

	$\bar{\bar{A}}\bar{B}\bar{C}$	$\bar{A}\bar{B}\bar{C}$	$A\bar{B}\bar{C}$	$\bar{A}B\bar{C}$
$\bar{\bar{C}}D$	$\bar{\bar{A}}\bar{B}\bar{C}D$	$\bar{A}\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$	$\bar{A}B\bar{C}D$
$\bar{C}D$	$\bar{\bar{A}}\bar{B}C D$	$\bar{A}\bar{B}C D$	$A\bar{B}C D$	$\bar{A}BC D$
$\bar{C}D$	$\bar{\bar{A}}\bar{B}\bar{C} \bar{D}$	$\bar{A}\bar{B}\bar{C} \bar{D}$	$A\bar{B}\bar{C} \bar{D}$	$\bar{A}B\bar{C} \bar{D}$
$\bar{C}D$	$\bar{\bar{A}}\bar{B}C \bar{D}$	$\bar{A}\bar{B}C \bar{D}$	$A\bar{B}C \bar{D}$	$\bar{A}BC \bar{D}$

Склеивание осуществляется между теми минтермами, которые записаны в виде "1" в соседних клет-

ках карты (по вертикали или горизонтали). Соседними считаются клетки крайнего левого и правого, верхнего и нижнего рядов (можно представить карту как развертку цилиндра). Два минтерма, находящиеся в соседних клетках можно представить в виде одного логического произведения переменных, число которых на одну единицу меньше, чем в каждом из соседних минтермов, причем в произведении остаются общие для обоих минтермов сомножители. Если соседними окажутся сразу 4 минтерма с "1" то такую группу можно заменить произведением переменных, число которых меньше, чем в каждом минтерме, уже на два. Учитывая, что $A + A + \dots + A = A$, одну

единицу, изображающую минтерм, можно объединять пары несколько раз.

Используя метод карт Карно минимизируем рассмотренную в примере функцию. Единица, изображающая минтерм ABC входит сра-

	$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	AB
\bar{C}	0	0	1	0
C	0	1	1	0

зу в два объединения, обозначенные

а и b. Объединение а отражает

"склеивание" минтермов ABC и $AB\bar{C}$

$$ABC + AB\bar{C} = AB (C + \bar{C}) = AB$$

объединение b отражает склеивание минтермов ABC и $\bar{A}BC$

$$ABC + \bar{A}BC = BC (A + \bar{A}) = BC$$

В результате проведенных операций "склеивания" из трех минтермов, входящих в функцию и являющихся конъюнкцией трех переменных, остались лишь слагаемые AB и BC.

Отсюда: $Y = AB + BC$.

Домашнее задание N2 по курсу "Электроника и МПТ".

Разработать цифровое устройство на основе любого микропроцессора, выполняющее функцию, минимизированную в домашнем задании N1 (Включая принципиальную схему и управляющую программу в кодах выбранного микропроцессора) .

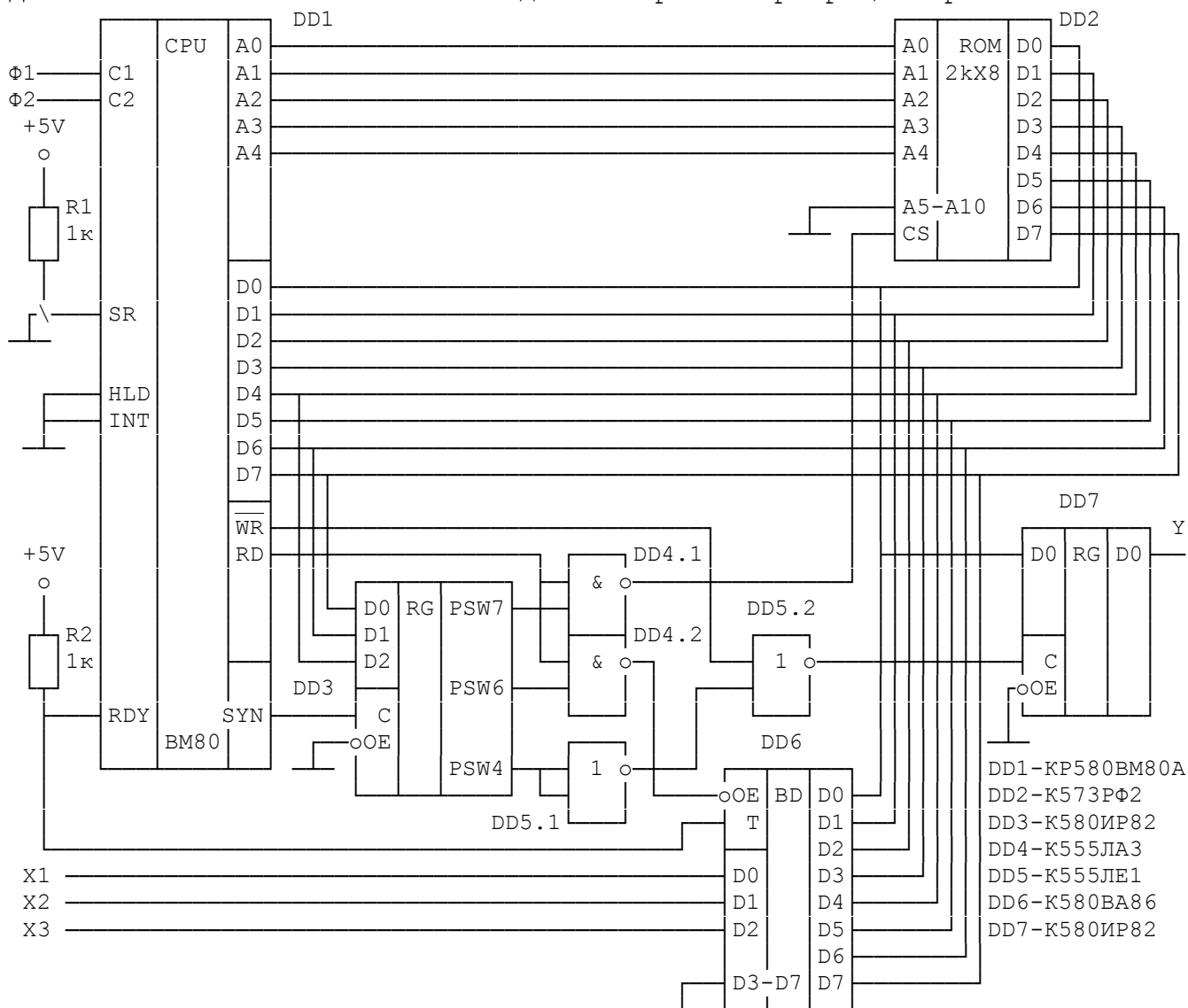
ПРИМЕР: Задана в виде таблицы истинности логическая функция Y трех переменных X_1, X_2, X_3 .

x3	x2	x1	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

$$Y = \overline{x_1} \overline{x_2} \overline{x_3} + x_1 \overline{x_2} \overline{x_3} + x_1 x_2 \overline{x_3} + x_1 x_2 x_3 =$$

$$= \overline{x_2} \overline{x_3} + x_1 x_2$$

Для выполнения поставленной задачи выбран микропроцессор КР580ВМ80А.



Пример построения МП – устройства на основе МП КР580ВМ80А

Рассмотрим принципиальную схему МП – системы, реализующей заданную функцию под управлением составленной программы (рис.).

Система построена на базе центрального процессора DD1 KP580BM80A, синхронизируемого сигналами Ф1 и Ф2 от внешнего генератора. Управляющая программа содержится в ПЗУ DD2. Ячейки ПЗУ адресуются по шести младшим линиям шины адреса микропроцессора, что вполне достаточно для выборки 40 байт программы. Выбранный в ПЗУ байт поступает в шину данных системы, образованную внешней шиной данных МП, выводами данных ПЗУ DD2, регистров DD3 и DD7 и шинного формирователя DD6.

Входные сигналы поступают по линиям X1, X2, и X3 на вход шинного формирователя DD6, образующего с элементом DD4.2 порт ввода. Выходной сигнал поступает на линию Y с выхода регистра DD7, образующего с элементами DD5 порт вывода.

В начале каждого цикла на шину данных МП выводится слово состояния процессора. В рассматриваемой системе используется только 3 разряда PSW – разряд, информирующий о начале цикла вывода данных в порт вывода (PSW4), разряд, информирующий о вводе из порта ввода (PSW6) и разряд, информирующий о чтении памяти (PSW7). Значения этих разрядов записываются в регистр слова состояния на основе регистра-защелки DD3. Запись синхронизируется сигналом SYN (СИНХРО). Эти сигналы позволяют отличать циклы обращения к памяти от циклов обращения к портам ввода – вывода. При выполнении процессором цикла чтения памяти в разряд PSW7 записывается логическая 1, в остальные разряды – нулевые значения. Сигнал PSW7 поступает на вход логического элемента DD4.1. На второй вход подается сигнал RC, информирующем о том, что процессор считывает сигналы с шины данных. Сигнал с выхода этого элемента подается на вход CS микросхемы ПЗУ. Единичный сигнал на CS переводит линии данных ПЗУ в z – состояние. При активном единичном сигнале RC на выходе DD4.1 появляется уровень логического 0, Z – состояние снимается и байт данных поступает из ПЗУ в МП.

В процессе выполнения цикла чтения из порта IN (port) в разряде PSW6 появляется 1. При активном сигнале RC на выходе элемента DD4.2 появляется лог. 0, который переводит шинный формирователь DD6 в режим передачи информации с входных линий X1, X2 и X3 на четыре младших разряда шины данных.

В процессе выполнения цикла записи в порт OUT (port) в разряде PSW4 появляется 1. При активном нулевом сигнале TR на выходе элемента DD5.2

появляется лог. 1, которая переводит регистр DD7 в режим записи информации с шины данных. Бит D0 с шины данных поступает на выходную линию Y и сохраняется на ней до следующего цикла вывода.

Так как в рассматриваемом устройстве используется только один порт ввода и один порт вывода, выборка их по адресам не требуется и, поэтому в программе адреса портов могут быть любыми.

Единичный сигнал SR производит сброс микропроцессора в исходное состояние. В этом случае в счетчик команд записывается 0, т.е. начинается выполнение программы с адреса 0000H.

Входные сигналы будем считывать через порт ввода с адресом 00H

0	0	0	0	0	X3	X2	X1
---	---	---	---	---	----	----	----

Выходные сигналы будем выводить через порт вывода с адресом 01H

0	0	0	0	0	0	0	Y
---	---	---	---	---	---	---	---

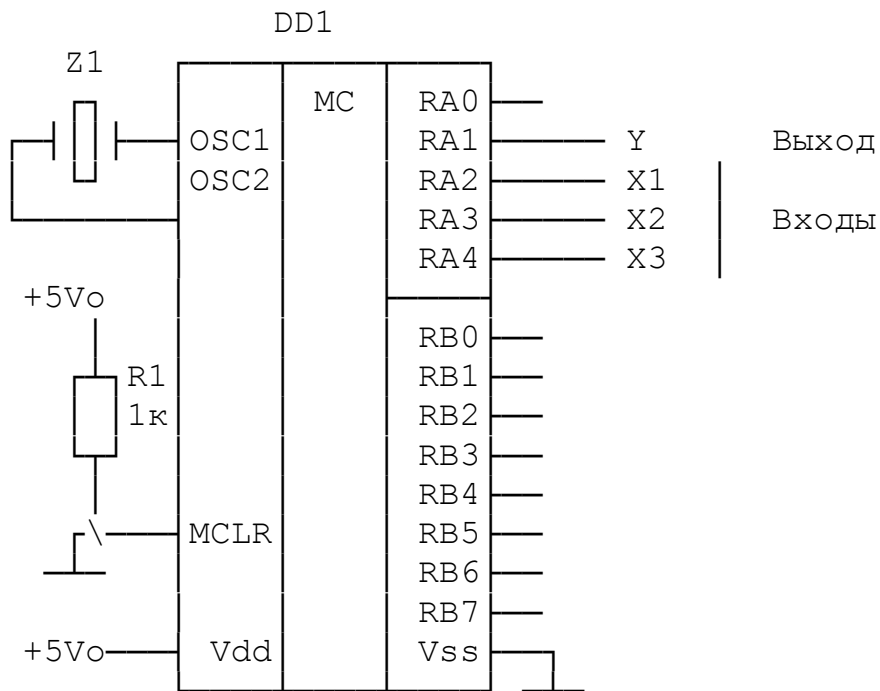
Действие	оператор	код	мт
1. Загрузить данные из порта ввода в A	IN 00H	D8 00	10
2. Сохранить слово данных в рег. E	MOV E,A	5F	5
3. Выделить X3: (A) AND 00000100 (Обнуляются все разряды, кроме 2)	ANI 04H	E6 04	7
4. Сдвиг (A) вправо	RRC	0F	4
5. Сдвиг (A) вправо	RRC	0F	4
6. Сохранить X3 в рег. B: B <- (A)	MOV B,A	47	5
7. Восстановить слово данных в A	MOV A,E	7B	5
8. Выделить X2: (A) AND 00000010 (Обнуляются все разряды, кроме 1)	ANI 02H	E6 02	7
9. Сдвиг (A) вправо	RRC	0F	4
10. Сохранить X2 в рег. C: C <- (A)	MOV C,A	4F	5
11. Восстановить слово данных в A	MOV A,E	7B	5
12. Выделить X1: (A) AND 00000001 (Обнуляются все разряды, кроме 0)	ANI 01H	E6 01	7
13. Произведение X1 X2: (A) /\ (C)	ANA C	A1	4
14. Сохранить X1 X2 в рег. D: D <- (A)	MOV D,A	57	5

ПРИМЕР: Задана в виде таблицы истинности логическая функция Y трех переменных x1, x2, x3.

x3	x2	x1	Y	В результате минимизации получена следующая функция:
0	0	0	1	
0	0	1	1	$\overline{x_2} \overline{x_3} + x_1 x_2$
0	1	0	0	
0	1	1	1	
1	0	0	0	
1	0	1	0	
1	1	0	0	
1	1	1	1	

Для выполнения поставленной задачи выбран микроконтроллер PIC16F84A.

Принципиальная схема содержит сам МК, цепь тактирования на основе кварцевого резонатора и цепь сброса.



Для обмена данными используется порт ввода/вывода RA.

Слово конфигурации МК выглядит следующим образом:

`__CONFIG __CP_OFF & __WDT_OFF & __PWRTE_ON & __XT_OSC`

`__CP_OFF` - отключается защита памяти от чтения;

`__WDT_OFF` - отключается сторожевой таймер;

`__PWRTE_ON` - включается таймер задержки сброса МК после подачи питания;

`__XT_OSC` - тактирование от кварцевого резонатора.

Программа вводит три сигнала через порт А (x1-RA4, x2-RA3, x3-RA2),
 вычисляет функцию трех логических переменных
 $y = x1 \wedge x2 \vee x2 \wedge x3$ и выводит результат в порт А (y-RA1).
 Следовательно, формат порта следующий:

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
Отсутствуют, читаются как 0			X3	X2	X1	Y	0

Действие	Команда	Адрес в памяти программ	Код команды	Время выполнения
---- инициализация контроллера ----				
устанавливается банк памяти 1	bcf STATUS,RP1	0000	1303	1
	bsf STATUS,RP0	0001	1683	1
настройка порта А: RA4, RA3, RA2 - на ввод, RA1, RA0 - на вывод	movlw 0x1C	0002	301C	1
	movwf TRISA	0003	0085	1
Настройка порта В: все линии на ввод	movlw 0xFF	0004	30FF	1
	movwf TRISB	0005	0086	1
возврат в банк 0	bcf STATUS,RP0	0006	1283	1
---- Основной исполнительный цикл ----				
----- получаем комбинацию переменных и раскладываем их по разным регистрам -----				
чтение данных из порта А в W	movf PORTA, 0	0007	0805	1
Сохранение в ячейке 0x0F	movwf 0x0F	0008	008F	1
наложение маски 00000100 - выделение x1	andlw 0x04	0009	3904	1
Сохранение x1 в ячейке 0x0E	movwf 0x0E	000A	008E	1
восстановление исходных данных в W	movf 0x0F, 0	000B		1
наложение маски 00001000 - выделение x2	andlw 0x08	000C	3908	1
сохранение x2 в ячейке 0x0D	movwf 0x0D	000D	008D	1
сдвиг x2 вправо-выравнивание с x1	rrf 0x0D, 1	000E	0C8D	1
восстановление исходных данных в W	movf 0x0F, 0	000F	080F	1
наложение маски 00010000 - выделение x3	andlw 0x10	0010	3910	1
Сохранение x3 в ячейке 0x0C	movwf 0x0C	0011	008C	1
сдвиг x3 вправо	rrf 0x0C, 1	0012	0C8C	1
сдвиг x3 вправо-выравнивание с x1	rrf 0x0C, 1	0013	0C8C	1
---- вычисление функции ----				
инверсия x3	comf 0x0C, 1	0014	098C	1
инверсия x2 с сохранением в W	comf 0x0D, 0	0015	090D	1
логическое умножение $x2 \wedge x3$ с сохранением в 0x0C	andwf 0x0C, 1	0016	058C	1
копирование x2 в W	movf 0x0D, 0	0017	080D	1
логическое умножение $x1 \wedge x2$ с сохранением в W	andwf 0x0E, 0	0018	050E	1
логическая сумма $x2 \wedge x3 + x1 \wedge x2$ с сохранением в 0x0C - РЕЗУЛЬТАТ	iorwf 0x0C, 1	0019	048C	1
сдвиг вправо - выравнивание с y	rrf 0x0C, 1	001A	0C8C	1
наложение маски 00000010 - выделение y перед выводом в порт	movlw 0x02	001B	3002	1
	andwf 0x0C, 1	001C	058C	1
----- вывод результата в порт по алгоритму "чтение - модификация - запись" -----				
чтение содержимого порта А	movf PORTA, 0	001D	0805	1
обнуление бита 01 в байте, считанном из порта	andlw 0xFD	001E	39FD	1
установка y в бит 01 с сохран. в W	iorwf 0x0C, 0	001F	040C	1
Вывод в порт А	movwf PORTA	0020	0085	1
Возврат на начало цикла	goto 0x0007	0021	2807	2

Итого 28 машинных циклов, т.е. при тактовой частоте 4 МГц время выполнения основного исполнительного цикла 28 мкс, код программы занял 34 ячейки памяти программ и использованы 4 ячейки памяти данных (ОЗУ).

После подачи напряжения питания устройство начинает работать и выполняется программа, записанная в память программ, начиная с адреса 0x0000. С адреса 0x0004 может начинаться подпрограмма обработки прерывания. Так как описываемое устройство не использует технологию прерываний, эта особенность не учитывается.

Вначале происходит назначение режимов работы портов ввода/вывода – определение направления передачи информации. Эта процедура выполняется единожды за весь сеанс работы. Обработка данных происходит в бесконечном цикле. Весь цикл можно условно разделить на три части: ввод информации и ее обработка; вычисление логической функции; вывод результата и возврат на начало цикла.

Листинг программы приведен в приложении.

ПРИЛОЖЕНИЕ

```
;*****
;   Filename:      example_dz-pic.asm
;   Программа вводит три сигнала через порт А (x1-RA2, x2-RA3, x3-RA4),
;   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

;***** VARIABLE DEFINITIONS
var1      EQU      0x0C   ;      x1
var2      EQU      0x0D   ;      x2      - входные переменные
var3      EQU      0x0E   ;      x3
datainp   EQU      0x0F
d1        EQU      0x10
d2        EQU      0x11
;*****

        ORG        0x000           ; processor reset vector
        bcf        STATUS,RP1      ; устанавливаем
        bsf        STATUS,RP0      ; банк памяти 1
        movlw      0x1C            ; управляющее слово для порта А
        movwf      TRISA           ; RA4, RA3, RA2 - на ввод, RA1, RA0 - на вывод
        movlw      0xFF            ; управляющее слово для порта В
        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,
; индицирующий работоспособность устройства
        movlw      0xff
        movwf      PORTA           ; погасили
        call       del_100         ; подождали 100 мс
        call       del_100         ; подождали 100 мс
        movlw      0x00
        movwf      PORTA           ; зажгли
        call       del_100         ; подождали 100 мс
        call       del_100         ; подождали 100 мс
        movlw      0xff
        movwf      PORTA           ; погасили
        call       del_100         ; подождали 100 мс
        call       del_100         ; подождали 100 мс
        movlw      0x00
        movwf      PORTA           ; зажгли
; -----
; ----- Основной исполнительный цикл -----
;   y=x2^x3+x1x2
```

```

; - получаем комбинацию переменных и раскладываем их по разным регистрам
; X X X x3 x2 x1 y 0 -- формат порта A

start    movf    PORTA, w      ; читаем данные из порта A
         movwf   datainpw     ; сохраняем
         andlw   0x04         ; накладываем маску 00000100 - выделяем x1
         movwf   var1         ; сохраняем в var1
         movf    datainpw, w   ; восстанавливаем исходные данные в аккумулятор
         andlw   0x08         ; накладываем маску 00001000 - выделяем x2
         movwf   var2         ; сохраняем в var2
         rrf     var2, f       ; сдвиг x2 вправо - выравниваем с x1
         movf    datainpw, w   ; восстанавливаем исходные данные в аккумулятор
         andlw   0x10         ; накладываем маску 00010000 - выделяем x3
         movwf   var3         ; сохраняем в var3
         rrf     var3, f       ; сдвиг x3 вправо
         rrf     var3, f       ; сдвиг x3 вправо - выравниваем с x1
; ----- вычисляем функцию -----
         comf    var3, f       ; инвертируем x3 с сохранением в var3
         comf    var2, w       ; инвертируем x2 с сохранением в аккумуляторе
         andwf   var3, f       ;  $x2 \wedge x3$  с сохранением в var3
         movf    var2, w       ; x2 - в аккумулятор
         andwf   var1, w       ;  $x1 \wedge x2$  с сохранением в аккумуляторе
         iorwf   var3, f       ;  $x2 \wedge x3 + x1 \wedge x2$  с сохранением в var3 - РЕЗУЛЬТАТ
         rrf     var3, f       ; сдвиг вправо - выравниваем с y с сохранением в var3
         movlw   0x02         ; накладываем маску 00000010 -
         andwf   var3, f       ; - чистим результат перед выводом в порт
; ----- выводим результат в порт по алгоритму "чтение - модификация - запись --
         movf    PORTA, w      ; читаем данные из порта A
         andlw   0xFD         ; обнуляем бит 1 в байте, считанном из порта
         iorwf   var3, w       ; устанавливаем y
         movwf   PORTA         ; выводим в порт

         goto    start        ; Зацикливаем программу
; ----- Итого 28 машинных циклов, т.е. при тактовой частоте 4 МГц время выполнения
цикла 28 мкс.

; Подпрограмма - задержка 100 мс
; Переменные: d1 и d2
del_100  movlw   0x66
         movwf   d2
d100_2   movlw   0xF9
         movwf   d1
         nop
         nop
d100_1   nop
         decfsz  d1, f
         goto    d100_1
         nop
         decfsz  d2, f
         goto    d100_2
         return
; ----- end of DEL_100 -----

end      ; directive 'end of program'

```