



**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И
СВЯЗИ ИМЕНИ
ГУМАРБЕКА ДАУКЕЕВА**

Кафедра
«Электроснабжение
и возобновляемые
источники энергии»

ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

Методические указания и задания по выполнению лабораторных работ для студентов специальности 5В071800 – «Электроэнергетика»

Алматы 2021

СОСТАВИТЕЛЬ: Акименков М.В. Основы цифровой техники. Методические указания и задания по выполнению лабораторных работ для студентов специальности 5В071800 – «Электроэнергетика» – Алматы: АУЭС, 2021, 16 с.

Представлены методические указания и задания к выполнению лабораторных работ по дисциплине «Основы цифровой техники 3208». Цель работ – изучение процесса разработки программ на Assembler для микроконтроллеров в On-line режиме с использованием программного комплекса MPLAB фирмы Microchip в режиме симулятора, что позволяет студентам самостоятельно выполнять лабораторные работы на своих ПК и ноутбуках без использования программатора.

Иллюстраций – 6, таблиц – 2, библиографий – 2 названия.

Рецензент: канд. пед. наук, доц. Саламатина А.М

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи имени Гумарбека Даукеева» на 2021 год.

© НАО «Алматинский университет энергетики и связи имени Гумарбека Даукеева», 2021 г.

Содержание

Введение.....	4
1. Лабораторная работа № 1 «Прямая и косвенная адресация».....	4
2. Лабораторная работа № 2 «Использование TMR0 в режиме запрещенного прерывания».....	7
3. Лабораторная работа № 3 «Использование TMR0 в режиме разрешенного прерывания».....	11
4. Лабораторная работа № 4 «Использование таймера TMR2»	15
Список литературы	21

Введение

Лабораторные работы по дисциплине «Основы цифровой техники» направлены на приобретение знаний по разработке программ для микроконтроллеров на языке Assembler. В современной электроэнергетике микроконтроллеры используются в SCADA, АСКУЭ, РЗА, на основе которых существует управление в этой отрасли, как и во многих других. Программное обеспечение компании Microchip Technology Incorporated, поставляемое компанией для общего пользования, позволяет студентам на своих ПК и ноутбуках в режимах дистанционного обучения приобрести первоначальные знания по разработке программ для МК, реализующих алгоритмы управления, выполняемые ими. Алгоритмы программ в лабораторных работах имитируют фрагменты технологических процессов автоматического управления в электроэнергетических системах.

1. Лабораторная работа № 1 «Прямая и косвенная адресация»

Косвенная адресация применяется в случае необходимости работы с изменением данных в массивах. В этом случае получается более компактная программа. Для выполнения косвенной адресации необходимо обратиться к физически не реализованному регистру INDF. Обращение к регистру INDF фактически вызовет действие с регистром, адрес которого указан в регистре FSR. 9-й бит косвенного адреса IRP сохраняется в 7-м бите регистра STATUS. Если $IRP = 0$, операции выполняются в банках 0, 1. Если $IRP = 1$, то в банках 2, 3.

В лабораторной работе приведен фрагмент программы, выполняющий суммирование данных (например, показаний электросчетчиков определенного региона), хранящихся в регистрах общего назначения банка 0. Суммирование выполняется методами косвенной и прямой адресации.

Таблица 1 – Варианты задания для лабораторной работы № 1

Данные	Номер варианта								
	1	2	3	4	5	6	7	8	9
Data1	2	3	4	5	6	7	8	9	10
Data2	3	3	3	3	3	3	3	3	3
Data3	4	4	4	4	4	4	4	4	4
Data4	5	5	5	5	5	5	5	5	5
Data5	6	6	6	6	6	6	6	6	6
sum1	10	11	12	13	14	15	16	17	18

Таблица 2 – Варианты задания для лабораторной работы № 1

Данные	Номер варианта								
	10	11	12	13	14	15	16	17	18
Data1	11	12	13	14	15	16	17	18	19
Data2	3	3	3	3	3	3	3	3	3
Data3	4	4	4	4	4	4	4	4	4
Data4	5	5	5	5	5	5	5	5	5
Data5	6	6	6	6	6	6	6	6	6
sum1	10	11	12	13	14	15	16	17	18

Таблица 3 – Варианты задания для лабораторной работы № 1

Данные	Номер варианта								
	19	20	21	22	23	24	25	26	27
Data1	20	21	22	23	24	25	26	27	28
Data2	3	3	3	3	3	3	3	3	3
Data3	4	4	4	4	4	4	4	4	4
Data4	5	5	5	5	5	5	5	5	5
Data5	6	6	6	6	6	6	6	6	6
sum1	10	11	12	13	14	15	16	17	18

В приведенном ниже тексте программы студент заменяет значения исходных данных sum1, Data1–5 в массиве по своему варианту. В Ndata заносится общее число данных – 5.

```
include <p16f877A.inc>
Data1 EQU H'22'
Data2 EQU H'23'
Data3 EQU H'24'
Data4 EQU H'25'
Data5 EQU H'26'
Ndata EQU H'27'
sum1 EQU h'21'
    org h'00'
    nop
    nop
    nop
    org h'05'
    clrf STATUS
    movlw d'1'
    movwf Data1
```

```

movlw d'2'
movwf Data2
movlw d'3'
movwf Data3
movlw d'4'
movwf Data4
movlw d'5'
movwf Data5
movlw d'5'
movlw d'10'
movwf sum1
movlw d'5'
movwf Ndata
bcf STATUS, IRP
movlw h'22'
movwf FSR
movf sum1,w
NEXT
addwf INDF,w
incf FSR,f
decfsz Ndata
goto NEXT
movwf sum1
pop
movlw d'10'; это прямая адресация
movwf sum1
movf sum1,w
addwf Data1,w
addwf Data2,w
addwf Data3,w
addwf Data4,w
addwf Data5,w
pop
end

```

Для контроля работы программы создается addwf INDF,w окно наблюдения, в которое необходимо вывести все переменные: sum1, Data1-5 и Ndata. Выполняется пошагово программа, снимается скриншот после выполнения всех операций суммирования, когда маркер установлен на инструкции pop.

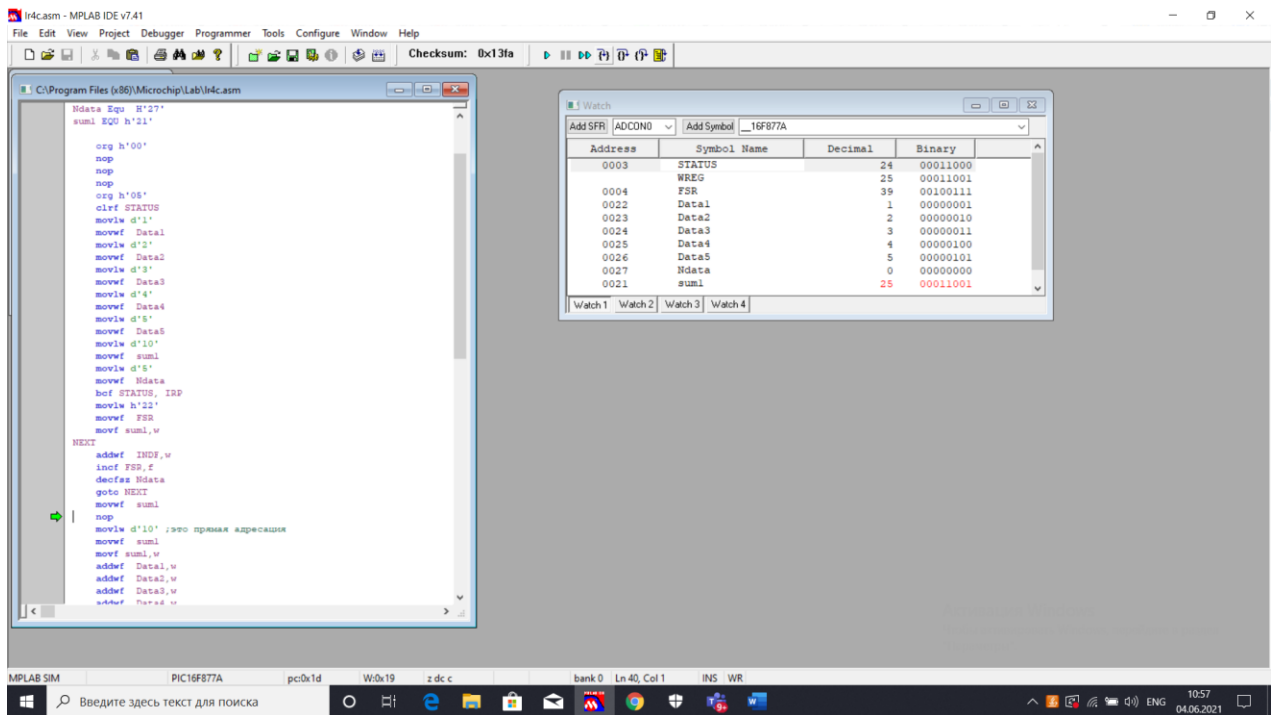


Рисунок 1 – Выполнено суммирование при косвенной адресации

Для выполнения операций суммирования при косвенной адресации требуется написать 8 инструкций, независимо от количества суммируемых данных. Количество инструкций при прямой адресации зависит от числа суммируемых данных. Если количество суммируемых данных обозначить через n , то надо написать $2n + 1$ инструкций.

Из данного примера видно, что с увеличением размерности массива применение косвенной адресации более предпочтительно.

Контрольные вопросы:

1. Что указывается в регистре FSR?
2. В каких банках находится регистр FSR?
3. В каких банках находится регистр INDF?
4. Что помещается в регистр INDF?
5. Где размещается 9-й бит косвенной адресации?
6. Где сохраняется результат выполнения операции `addwf INDF,w`?

2. Лабораторная работа № 2 «Использование TMR0 в режиме запрещенного прерывания»

Периферийный модуль TMR0 запускается после установки заданий в регистр OPTION_REG. Отсутствует настройка на его остановку. Обычно он используется в программах, где постоянно нужно контролировать длительность какого-то процесса по анализу появления флага от TMR0,

свидетельствующего о завершении цикла заполнения регистра TMR0. В данной лабораторной работе исследуется программа, которая предназначена для циклического включения и отключения некоторого технологического процесса. Сигналы на включение и отключение в виде напряжения 5 В подаются с определенных разрядов PORTC. Время цикла задается при настройке предделителя. В приведенной ниже таблице приведены значения бит 2–0 в регистре OPTION_REG, обеспечивающие настройку предделителя на замедление работы TMR0.

Таблица 4 – Значение бит 2–0 в регистре OPTION_REG

Биты 2–0	Коэффициент предделителя
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Таблица 5 – Варианты задания для лабораторной работы № 2

Исходные данные	Варианты задания								
	1	2	3	4	5	6	7	8	9
Бит включения	1	2	3	4	5	6	7	6	5
Бит отключения	2	1	2	3	4	5	6	7	6
Предделитель	1:4	1:8	1:16	1:32	1:64	1:128	1:256	1:4	1:8

Таблица 6 – Варианты задания для лабораторной работы № 2

Исходные данные	Варианты задания								
	10	11	12	13	14	15	16	17	18
Бит включения	2	1	2	3	4	5	6	7	6
Бит отключения	1	2	3	4	5	6	7	6	5
Предделитель	1:256	1:128	1:64	1:32	1:16	1:8	1:256	1:2	1:16

Таблица 7 – Варианты задания для лабораторной работы № 2

Исходные данные	Варианты задания								
	19	20	21	22	23	24	25	26	27
Бит включения	2	2	4	5	6	5	4	7	7
Бит отключения	3	4	3	4	5	7	7	2	5
Предделитель	1:256	1:128	1:64	1:32	1:16	1:8	1:256	1:2	1:16

Выполнение лабораторной работы

Создается проект по приведенному тексту программы с установкой по варианту задания бит включения и отключения, а также значение предделителя. В проекте создается окно наблюдения Watch, с включением PCH и POH согласно рисунку 2, и окно измерения времени Stopwatch.

```
include<p16f877a.inc>
org h'00'
nop
nop
nop
org h'05'
CLRF STATUS
CLRF PORTC; Очистка регистра PORTC
bcf INTCON,5; запрещаем прерывание от переполнения TMR0
bsf STATUS,5
clrf TRISC
movlw B'10000000'; Установка предделителя
; подтягивающие резисторы отключены, бит 7 = 0
; внешнее прерывание по заднему фронту сигнала, бит 6 = 0
; внутренний тактовый сигнал для TMR0, бит 5 = 0
; приращение TMR0 от внешнего тактового сигнала, бит 4 = 0
; предделитель включен перед TMR0, бит 3 = 0
; коэф. предделителя 1:2, биты 0–2 = 000
movwf OPTION_REG; TMR0 запущен
bcf STATUS,5
Main1
btfss INTCON,T0IF; Ждать переполнения TMR0 по появлению флага
goto Main1
bcf INTCON,T0IF; Сбросить флаг прерывания от TMR0
bsf PORTC,0
Main2
btfss INTCON,T0IF; Ждать переполнения TMR0 по появлению флага
goto Main2
bcf INTCON,T0IF; Сбросить флаг прерывания от TMR0
bcf PORTC,0
bsf PORTC,1
goto Main1
end
```

Устанавливаются BreakPoint, руководствуясь приведенным ниже скриншотом. Измеряется время длительности сигнала на включение и

отключение и снимаются скриншоты в местах, указанных в методических указаниях.

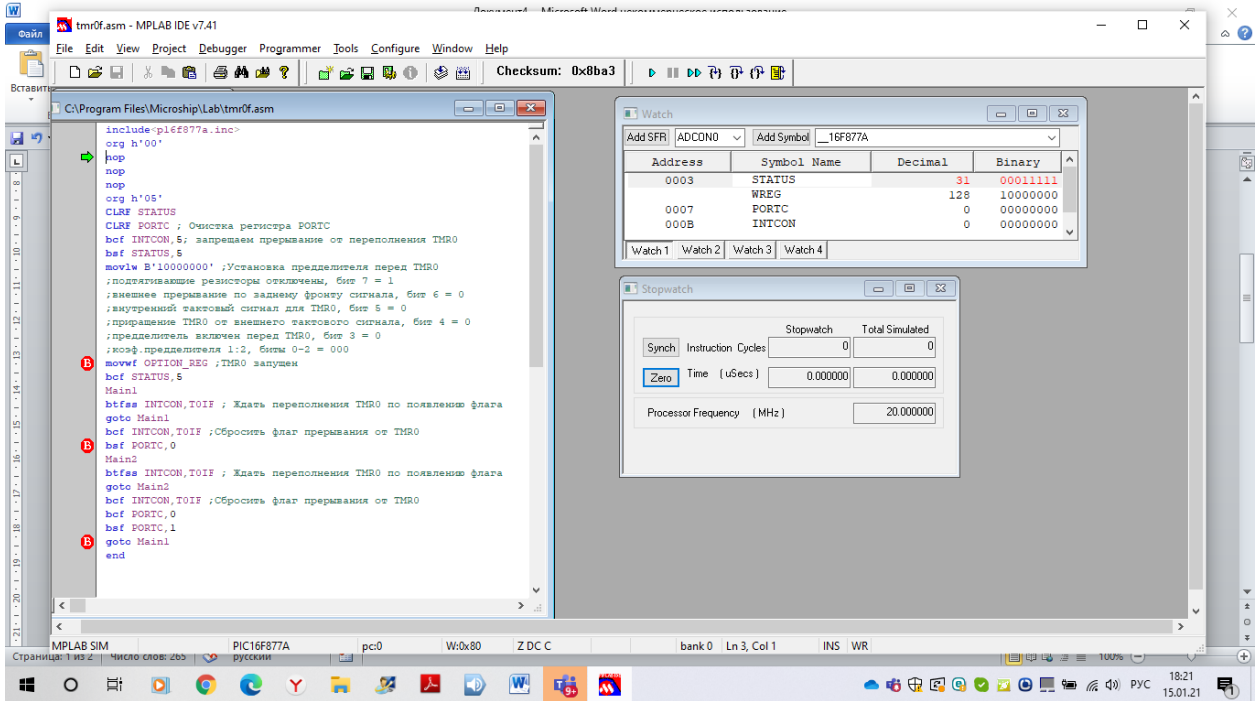


Рисунок 2 – Установка BreakPoint

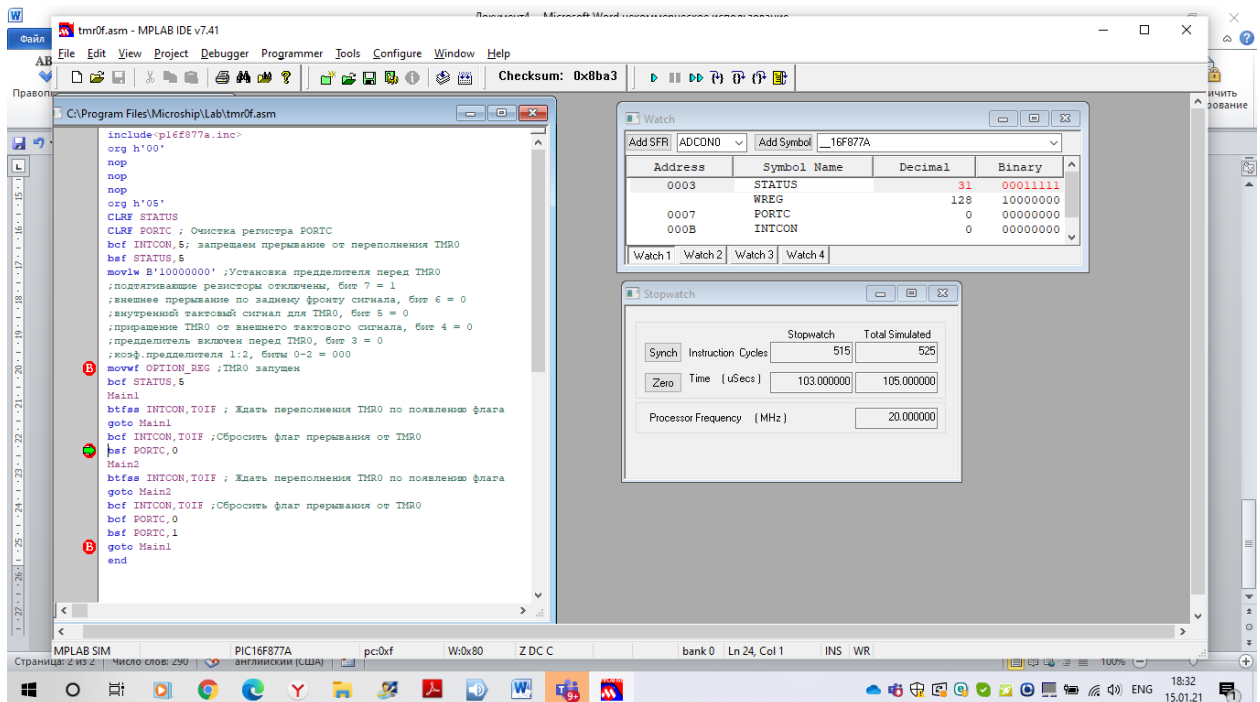


Рисунок 3 – Время отключенного процесса 103 мкс

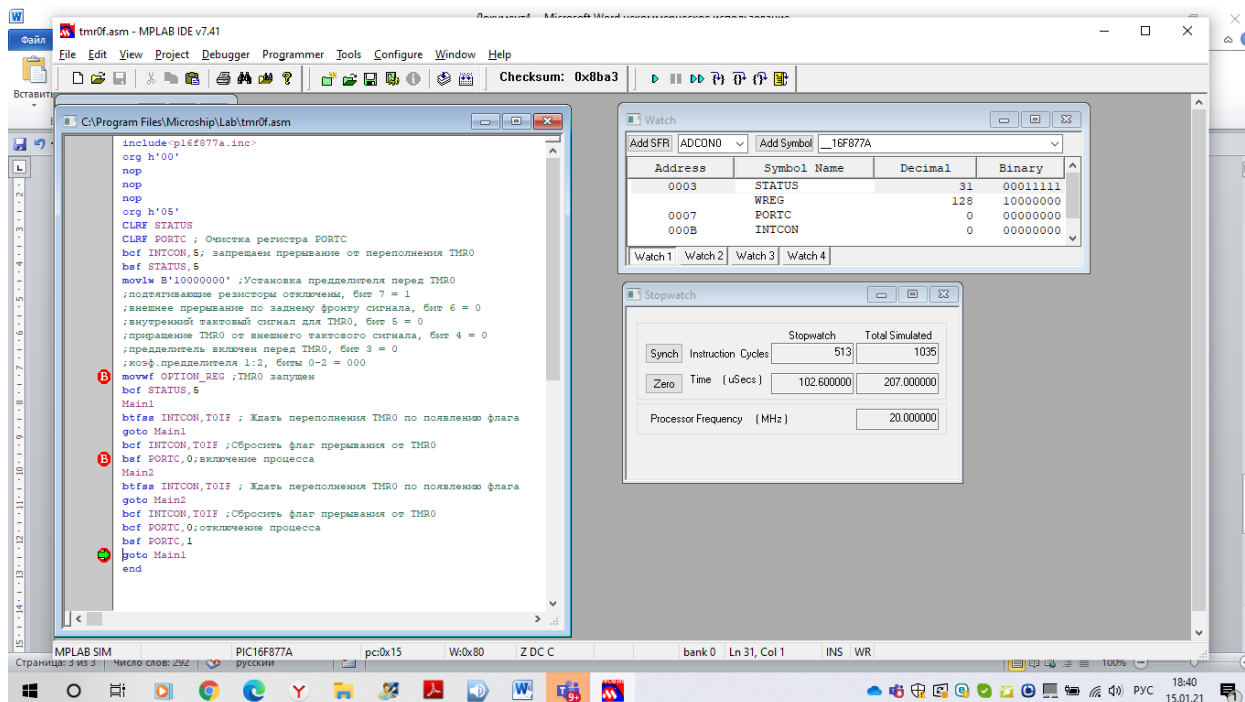


Рисунок 4 – Время включенного процесса 102,6 мкс

Контрольные вопросы:

1. Как установить запрет прерывания при переполнении TMR0?
2. В каком регистре появляется флаг, свидетельствующий о переполнении TMR0?
3. Можно ли в процессе запущенного TMR0 изменить значение делителя?
4. Можно ли прочитать установленное значение делителя?
5. Какой регистр используется для настройки TMR0?

Лабораторная работа № 3 «Использование TMR0 в режиме разрешенного прерывания»

Некоторые технологические процессы через определенные интервалы времени требуют выполнения определенного набора проверочных и настроечных операций, для выполнения которых требуется специальная программа. В этом случае целесообразно разрешить прерывание от TMR0 и оформить эту программу как программу обработки прерывания. В данной лабораторной работе приведен текст программы, реализующий этот принцип. Исходные данные в вариантах задания используются из предыдущей лабораторной работы. Программа обработки прерывания через определенный интервал времени, формируемый настройкой TMR0, должна переключать технологический процесс.

Выполнение лабораторной работы

Создайте проект по приведенному тексту программы, предварительно установив по варианту задания биты включения и отключения, а также значение предделителя.

```
Include<p16F877A.inc>
Cik1 EQU h'22'
Cik2 EQU h'23'
    ORG h'00'
    GOTO GLAV; при запуске программы сразу происходит переход
; на выполнение основной программы с пропуском инструкции
; перехода на подпрограмму обработки прерывания Prer
    NOP
    NOP
    NOP
    ORG h'04'
    CALL Prer
    NOP
GLAV
    NOP
    NOP
    CLRF STATUS
    CLRF PORTC; Очистка регистра PORTC
    CLRF Cik1
    CLRF Cik2
    movlw B'11100000'
    movwf INTCON; разрешаем все прерывания, прерывания от
; периферийных модулей и прерывание при переполнении TMR0
    bsf STATUS,5
    clrf TRISC; порт PORTC настроен на вывод
    movlw B'10000000'; Установка предделителя TMR0
; подтягивающие резисторы отключены, бит 7 = 1
; внешнее прерывание по заднему фронту сигнала, бит 6 = 0
; внутренний тактовый сигнал для TMR0, бит 5 = 0
; приращение TMR0 от внешнего тактового сигнала, бит 4 = 0
; предделитель включен перед TMR0, бит 3 = 0
; коэф.предделителя 1:2, биты 0-2 = 000
    movwf OPTION_REG; TMR0 запущен
    bcf STATUS,5
Prog
    bcf INTCON,T0IF; Сбросить флаг прерывания от TMR0
; Здесь может работать основная программа, которая
; будет прерываться через интервалы времени, формируемые
```

```

; TMR0 для перенастройки, выполняемые в программе
; обработки прерывания
btfss Clk2,0
goto Prog
goto $; далее подпрограмма обработки прерывания
Prer
    btfss Clk1,0
    goto Off
    goto On
Off bcf PORTC,0
    bsf PORTC,1
    bsf Clk1,0
    goto con1
On
    bsf PORTC,0
    bcf PORTC,1
    bcf Clk1,0
    goto con2

con1
    nop
con2
    nop
    retfie
end

```

Установите BreakPoint, руководствуясь приведенными ниже скриншотами. Измерьте время длительности сигнала на включение и отключение и снимите скриншоты. Для данного примера они приведены ниже.

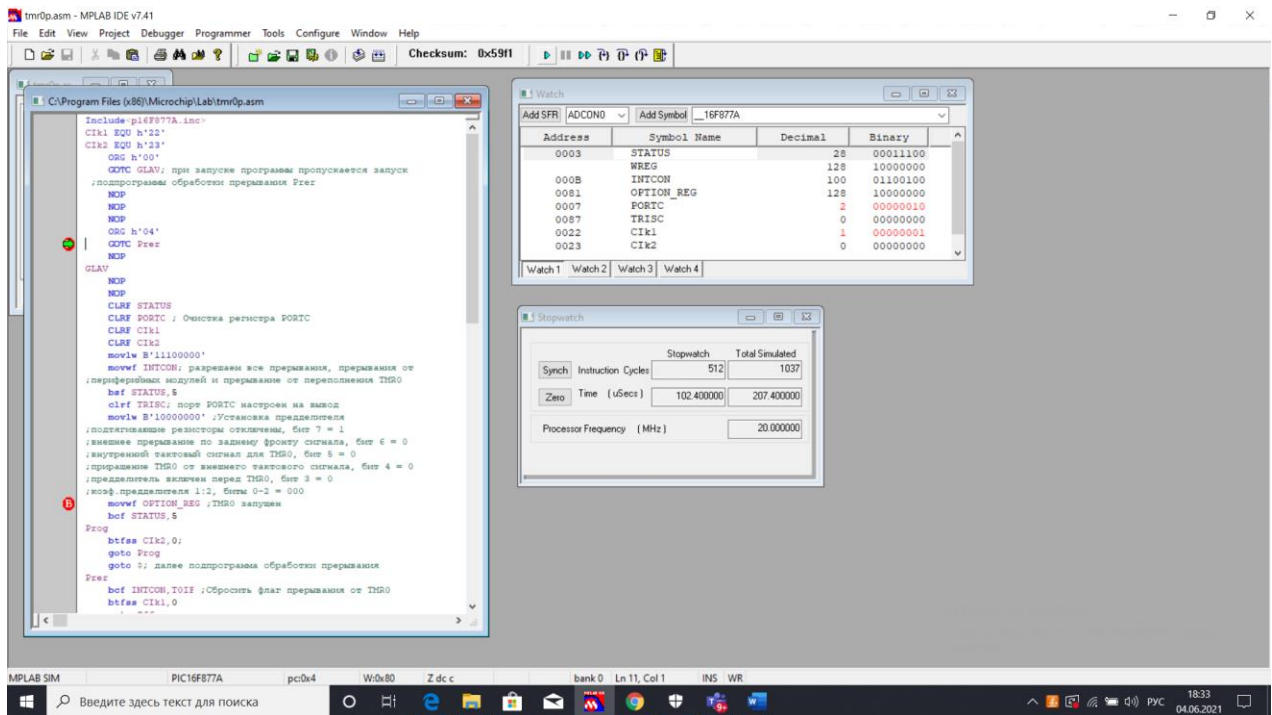


Рисунок 5 – Состояние 1

PORTC = 00000010, C1k1 = 00000001, C1k2 = 00000000, время работы процесса в первом состоянии 102,4 мкс.

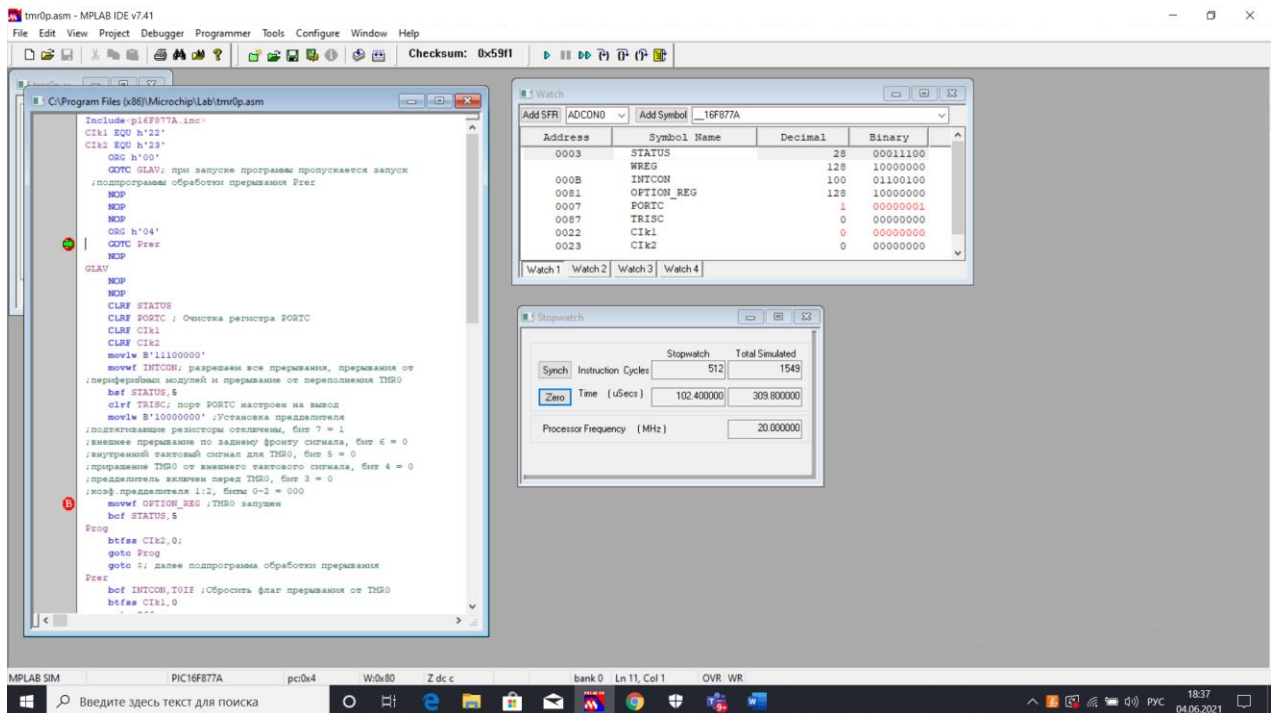


Рисунок 6 – Состояние 2

PORTC = 00000001, C1k1 = 00000000, C1k2 = 00000000, время работы процесса в состоянии 2 – 102,4 мкс.

На этих скриншотах мы видим, что переключение управления на Off и On с изменением состояния бит в порту PORTC<1,0> происходит через

интервалы длительностью 102,4 мкс. В одном состоянии бит 0 = 1, а бит 1 = 0. В другом состоянии бит 0 = 0, а бит 1 = 1.

Контрольные вопросы:

1. Как установить разрешение прерывания при переполнении TMR0?
2. В каком регистре появляется флаг, свидетельствующий о переполнении TMR0?
3. Какая переменная используется для организации переключения в технологическом процессе?
4. Можно ли прочитать установленное значение предделителя TMR0?
5. Какой регистр используется для настройки TMR0?
6. Для чего предназначается переменная Clk2?

Лабораторная работа № 4 «Использование таймера TMR2»

TMR2 – 8-разрядный таймер с программируемым предделителем и выходным делителем, 8-разрядным регистром периода PR2. Регистры TMR2 доступны для записи/чтения и очищаются при любом виде сброса.

Входной тактовый сигнал ($F_{osc}/4$) поступает через предделитель с программируемым коэффициентом деления (1:1, 1:4 или 1:16), определяемый битами T2CKPS1:T2CKPS0 (T2CON<1:0>).

TMR2 считает, инкрементируя от 00h до значения в регистре PR2, а затем сбрасывается в 00h на следующем машинном цикле. Регистр PR2 доступен для записи и чтения. После сброса значение регистра PR2 равно FFh.

Сигнал переполнения TMR2 проходит через выходной 4-разрядный делитель с программируемым коэффициентом деления (от 1:1 до 1:16 включительно) для установки флага TMR2IF в регистре PIR1<1>.

Для уменьшения энергопотребления таймер TMR2 может быть выключен сбросом бита TMR2ON (T2CON<2>) в 0. TMR2 может выступать опорным таймером для модуля ССР в режиме ШИМ.

TMR2 обладает широкими возможностями изменения по ходу выполнения программы режимов настройки длительности паузы. В данной программе используем изменение настройки выходного делителя, предделителя и задание для регистра PR2.

Таблица 8 – Коэффициент выходного делителя TMR2

T2CON <6-3>	Деление	T2CON <6-3>	Деление	T2CON <6-3>	Деление	T2CON <6-3>	Деление
0000	1:1	0100	1:5	1000	1:9	1100	1:13
0001	1:2	0101	1:6	1001	1:10	1101	1:14
0010	1:3	0110	1:7	1010	1:11	1110	1:15
0011	1:4	0111	1:8	1011	1:12	1111	1:16

Таблица 9 – Коэффициент деления предделителя TMR2

T2CON <1-0>	Деление
00	1:1
01	1:4
10	1:16

Таблица 10 – Задания на выполнение лабораторной работы № 4

Исходные данные	№ варианта							
	1	2	3	4	5	6	7	8
Выходной делитель 1	1:2	1:4	1:6	1:8	1:1	1:12	1:14	1:16
Выходной делитель 2	1:4	1:8	1:12	1:16	1:2	1:6	1:7	1:1
Регистр PR2	100	100	100	100	100	150	200	255
Предделитель 1	1:1	1:1	1:1	1:1	1:1	1:4	1:4	1:16
Предделитель 2	1:4	1:4	1:4	1:4	1:4	1:1	1:1	1:1

Таблица 11 – Задания на выполнение лабораторной работы № 4

Исходные данные	№ варианта							
	9	10	11	12	13	14	15	16
Выходной делитель 1	1:2	1:4	1:6	1:8	1:1	1:12	1:14	1:16
Выходной делитель 2	1:4	1:8	1:12	1:16	1:2	1:6	1:7	1:1
Регистр PR2	250	250	250	250	250	250	250	250
Предделитель 1	1:1	1:1	1:1	1:1	1:1	1:4	1:4	1:16
Предделитель 2	1:4	1:4	1:4	1:4	1:4	1:1	1:1	1:1

Таблица 12 – Задания на выполнение лабораторной работы № 4

Исходные данные	№ варианта							
	17	18	19	20	21	22	23	24
Выходной делитель 1	1:2	1:4	1:6	1:8	1:1	1:12	1:14	1:16
Выходной делитель 2	1:4	1:8	1:12	1:16	1:2	1:6	1:7	1:1
Регистр PR2	100	105	110	115	120	125	130	135
Предделитель 1	1:1	1:1	1:1	1:1	1:1	1:4	1:4	1:16
Предделитель 2	1:4	1:4	1:4	1:4	1:4	1:1	1:1	1:1

Пример программы, который используется студентами со своими данными из задания.

Таблица 13 – Задания на выполнение лабораторной работы № 4
для примера

Исходные данные	
Выходной делитель 1	1:16
Выходной делитель 2	1:1
Регистр PR2	255
Предделитель 1	1:16
Предделитель 2	1:1

```

include<p16f877a.inc>
org h'00'
nop
nop
nop
org h'05'
CLRF STATUS
CLRF PORTC; Очистка регистра PORTC
nach
bsf STATUS,5
bcf PIE1,1; запрещаем прерывание от переполнения TMR2
movlw b'11111111'; заносим в регистр PR2 число 255
movwf PR2
clrf TRISC
bcf STATUS,5
movlw B'01111110' ; Установка и включение TMR2
; выбор коэффициента выходного делителя 1:16, бит 6–3 = 1111
; включение модуля TMR2, бит 2 = 1
; выбор коэффициента предделителя 1:16, бит 1–0 = 10
movwf T2CON; TMR2 запущен
Main1
btfss PIR1,1; Ждать переполнения TMR2 по появлению флага
goto Main1
bcf PIR1,1; Сбросить флаг прерывания от TMR2
movlw B'00000100'; Установка и включение TMR2
; выбор коэффициента выходного делителя 1:1, бит 6-3 = 0000
; включение модуля TMR2, бит 2 = 1
; выбор коэффициента предделителя 1:1, бит 1–0 = 00
movwf T2CON; TMR2 работает с новыми установками
Main2
btfss PIR1,1; Ждать переполнения TMR2 по появлению флага
goto Main2
bcf PIR1,1; Сбросить флаг прерывания от TMR2

```

```

bsf STATUS,5
movlw b'00110011'; Заносим число в 5 раз меньшее (51) в регистр PR2
movwf PR2
bcf STATUS,5
movlw B'00000100'; Установка и включение TMR2 с новым PR2
; выбор коэффициента выходного делителя 1:1, бит 6-3 = 0000
; включение модуля TMR2, бит 2 = 1
; выбор коэффициента предделителя 1:1, бит 1-0 = 00
movwf T2CON ;TMR2 работает с PR2 в 5 раз меньшим
Main3
btfss PIR1,1; Ждать переполнения TMR2 по появлению флага
goto Main3
bcf PIR1,1; Сбросить флаг прерывания от TMR2
movlw B'00000000';
movwf T2CON; Останов TMR2
; отключение модуля TMR2, бит 2 = 0
goto nach
end

```

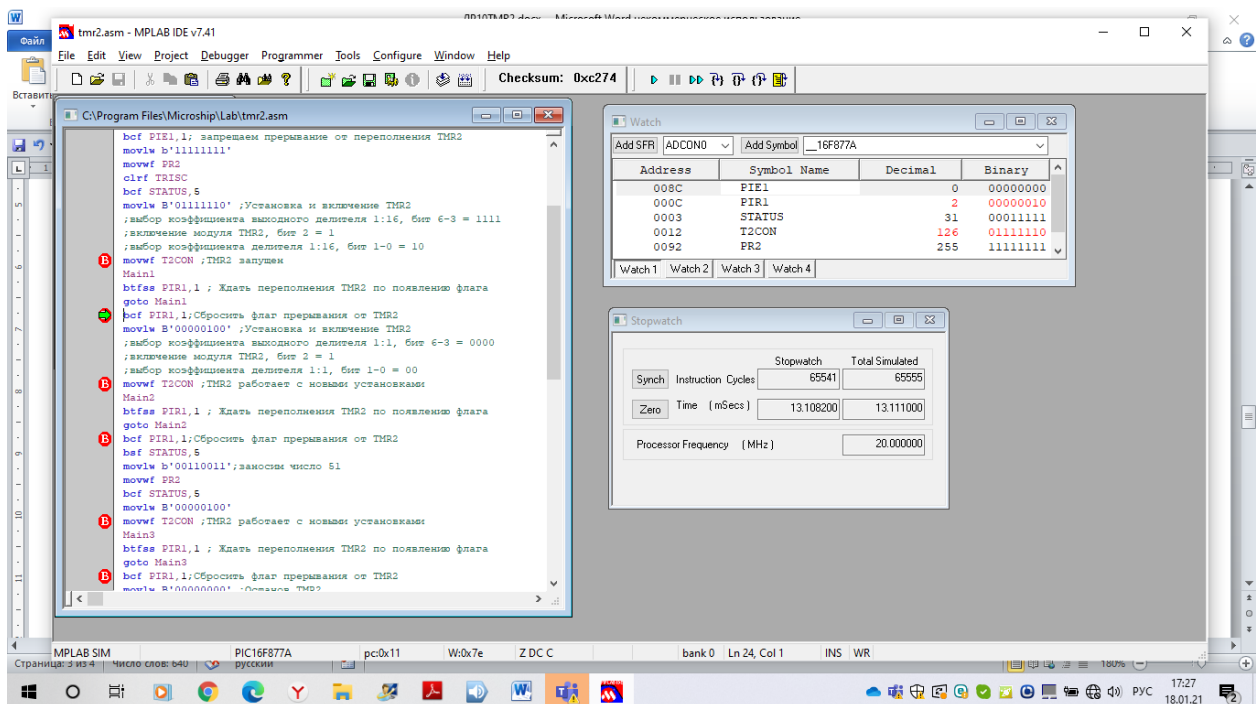


Рисунок 7 – Скриншот с первыми установками. Пауза составила 65541 цикл или 13,1082 мс

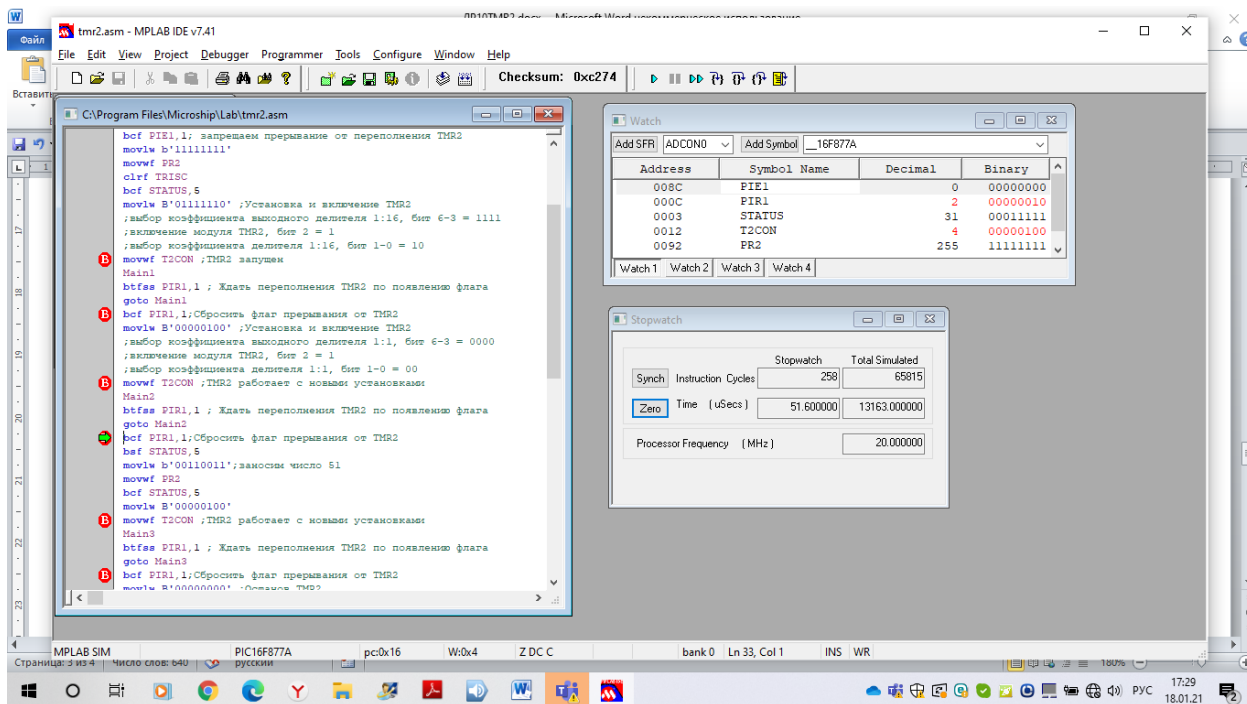


Рисунок 8 – Скриншот со вторыми установками. Пауза составила 258 циклов или 51,6 мкс.

Деление количества максимальных циклов на 256 дает значение 256,019. В регистр PR2 было занесено 255.

Теперь уменьшим в 5 раз значение, заносимое в PR2 – 51 и измерим время паузы.

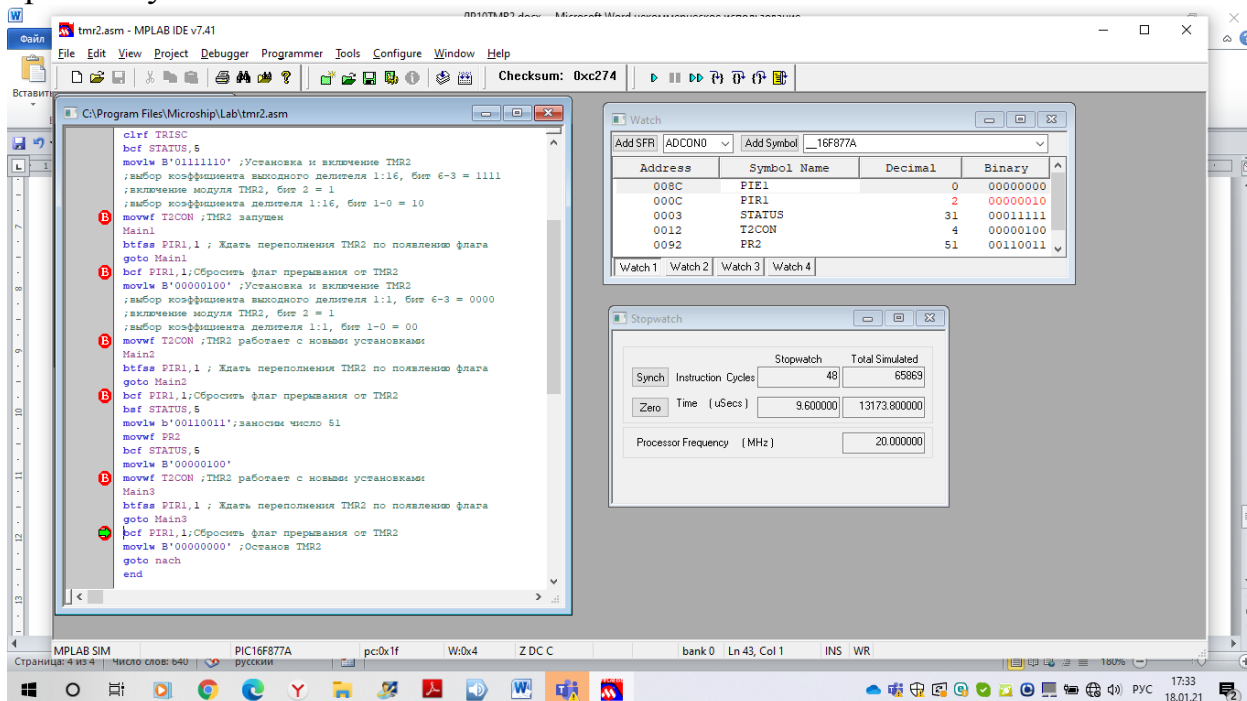


Рисунок 9 – С третьими установками пауза составила 48 циклов или 9,6 мкс.

Деление количества циклов во второй установке 258 на 5 дает значение 51,6. В регистр PR2 было занесено 51.

Анализ проведенных изменений в настройках TMR2, выполненных в процессе работы программы показывает:

- точное выставление времени задержки при настройке TMR2 можно определить только после проверки с использованием программы StopWatch;
- для изменения значения задания для PR2 необходимо переходить в банк 1. Это затрудняет изменение настроек TMR2 при перестройке PR2.
- наличие 3-х видов изменения настроек TMR2 позволяет более точно настроить требуемое время задержки.

Контрольные вопросы:

1. В каких пределах может быть настроен делитель TMR2?
2. На что влияет изменение настройки выходного делителя TMR2?
3. В каких пределах может быть настроен выходной делитель TMR2?
4. Для чего предназначен регистр PR2?
5. В каком банке находится регистр PR2?
6. Что надо сделать, чтобы в процессе работы программы изменить задание для PR2?

Список литературы

1. Однокристальные 8-разрядные FLASH CMOS микроконтроллеры компании Microchip Technology Incorporated. ООО «Микро-Чип». – Москва, 2002.
2. Акименков М.В. Методические указания по выполнению лабораторных работ по курсу «Основы цифровой (микропроцессорной) техники. – А.: АУЭС, 2015.
3. Сайт в Internet www.microchip.ru.
4. Катцен С. PIC микроконтроллеры. Все, что необходимо вам знать. – М.: Додека, 2008.
5. Копесбаева А.А. Микропроцессорные комплексы в системах управления. Учебное пособие. – Алматы: АИЭС, 2010.
6. Разработка программ на Assembler:
https://go.mail.ru/redirect?type=sr&redirect=eJzLKckpKLbS1y8qTSst1i0qLdYrKtUvSqwqSkzKL8lO1C0oyk8vSszN1U3JyUrUzc3MLsrPzs8rKcrPyUktyi_TZ2AwNDMyMrI0MTU3YrDm2L3b0jJk-8SQmolbPr_2BQBhKSMV&src=2a07f80&via_page=1&user_type=44&oid=2c3baac6b517c5ce
7. Создание схем и программирование для микроконтроллеров на Assembler:
https://go.mail.ru/redirect?type=sr&redirect=eJzLKckpKLbS1y8vL9erzC8tKU1K1UvOz9UvyEmszMksLrEHEbYBPp6-eZVJxt5lfoZZBh7G7hZGVVkhZkEG3j7u5ub-fmXpDAyGZkZGRpYmpuZGDPpp837eWxumLBrzZEBYxuq9ABTeI4A&src=544e5a0&via_page=1&user_type=44&oid=2c3baac6b517c5ce
8. Программирование микроконтроллеров:
https://go.mail.ru/redirect?type=sr&redirect=eJzLKckpKLbS1y8uKU0pSk3TS87P1TezNDA2sNDPzEvLL8pNLMnMTtQvKMpPL0rMzc0syi9LzMtMjc_NzC7Kz87PKynKz8lJB YoyMBiaGRkZWZqYmhsxXL_-gik-TEZqkcwSQ-GIEXsBdq4k8Q&src=371962e&via_page=1&user_type=44&oid=2c3baac6b517c5ce

Акименков Михаил Вениаминович

ОСНОВЫ ЦИФРОВОЙ ТЕХНИКИ

Методические указания и задания по выполнению лабораторных работ
для студентов специальности 5В071800 – «Электроэнергетика»

Редактор: Жанабаева Е.Б.

Специалист по стандартизации: Ануарбек Ж.А.

Подписано в печать _____

Тираж _20_ экз.

Объем 1,4 уч.- изд. л.

Формат 60x84 1/16

Бумага типографская №1

Заказ ____ Цена 650 тг.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи имени Гумарбека Даукеева»
050013, Алматы, ул. Байтурсынова, 126