



ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЕГО (ПОЛНОГО) ОБЩЕГО ОБРАЗОВАНИЯ

ЛИЦЕЙ ПРИ СПБГУТ

Вендор-ориентированный учебный курс в системе
«Старшая профильно-профессиональная школа-ВУЗ-Работодатель»:
«Программирование микроконтроллеров Microchip»

Богураев М.В., Кисляков С.В.

«УСЛОВНЫЕ ПЕРЕХОДЫ»

Методические указания к выполнению лабораторной работы

Санкт - Петербург
2009

Богураев М.В., Кисляков С.В. «УСЛОВНЫЕ ПЕРЕХОДЫ». Методические указания к выполнению лабораторной работы №3(5). СПб: ГОУ «Лицей при СПбГУТ», 2009.

ЛАБОРАТОРНАЯ РАБОТА №3 «УСЛОВНЫЕ ПЕРЕХОДЫ»

Цель работы

Научиться реализовывать ветвления в программах на ассемблере. Освоить моделирование в MPLAB IDE. Овладеть навыками программирования PIC микроконтроллеров.

Теоретические основы

Часто возникает необходимость сравнивать числа. Например, производится опрос устройства ввода и нужно выполнить некоторую подпрограмму по результатам опроса. Пусть устройство ввода – кнопка; и если кнопка не нажата, то на PORTB имеется число 0x0F. Если кнопка нажата, то на PORTB имеется число 0x0E. Для того чтобы определить нажата кнопка или нет, нужно сравнить содержимое ячейки памяти PORTB с числом 0x0F и 0x0E. В микроконтроллере PIC16F877A равенство или неравенство двух чисел можно определить с использованием операции вычитания. Если из одного числа вычесть другое и результат будет равен нулю, то числа равны. Если числа не были равны, то результат вычитания не может быть нулём. С учётом архитектуры нашего микроконтроллера код программы для определения равенства или неравенства двух чисел будет одинаков. Такой код будет содержать команду вычитания и проверку флага Z в регистре STATUS.

В математике операция вычитания записывается следующим образом:

$$a - b = c,$$

где a – уменьшаемое, b – вычитаемое, c – разность. Возможны три варианта после выполнения операции вычитания.

Если уменьшаемое равно вычитаемому: в этом случае разность будет равна нулю (В регистре STATUS флаг C = 1, флаг Z = 1).

Если уменьшаемое больше вычитаемого: в этом случае разность будет больше нуля (В регистре STATUS флаг C = 1, флаг Z = 0).

Если уменьшаемое меньше вычитаемого: в этом случае разность будет меньше нуля (В регистре STATUS флаг C = 0, флаг Z = 0).

В таблице 1 рассмотрены способы записи для сравнения чисел, находящихся в ячейках памяти, например, в ячейках a и b .

Табл. 1. Способы записи сравнения чисел

1	$a=b$	$b=a$
	$a \neq b$ (а не равно b)	$b \neq a$ (b не равно a)
2	$a > b$	$b < a$
	$a < b$	$b > a$
3	$a=0$	$b=0$

Для определения равенства нулю содержимого ячейки памяти, в нашем микроконтроллере достаточно выполнить запись ячейки в эту же ячейку. Это действие выполняется по команде MOVF f,F. Если содержимое ячейки памяти с названием f было нулевым, то выставляется флаг Z в регистре STATUS.

Практически все программы, выполняемые микроконтроллером, содержат фрагменты принятия решений, по результатам которых делается ветвление и выполняется одна или другая часть программы. Для принятия решений используют команды DECFSZ, INCFSZ, BTFSC, BTFSS. Для реализации ветвлений используют команды GOTO, CALL, RETFIE, RETLW, RETURN. На языке ассемблера условный переход всегда содержит команду условного перехода и одну или две команды безусловных переходов.

В этой работе для организации условного перехода одно число будет вычитаться из другого командой SUBWF. Для принятия решения командой BTFSS STATUS,Z будет

проверяться состояние бита Z в регистре STATUS. По результатам сравнения будет пропускаться (или не пропускаться) следующая команда безусловного перехода на другой фрагмент программы.

Регистр STATUS очень важен, поэтому отображается во всех четырёх банках памяти данных, и располагается в области регистров специальных функций (SFR) по адресам 03h, 83h, 103h, 183h. Нас интересует флаг (специальный бит) состояния арифметического логического устройства (АЛУ) микроконтроллера. Это флаг нулевого результата Z (второй бит в STATUS). Флаг Z устанавливается равным 1, если результат выполнения арифметической или логической операции ноль; флаг Z сбрасывается в 0, если результат выполнения арифметической или логической операции отличен от нуля.

По команде SUBWF содержимое регистра W вычитается из содержимого регистра F. То есть: регистр является уменьшаемым, а W вычитаемым. Регистр F имеет название, которое определяется директивой equ. Например: COUNTER equ 0x20. Тогда команда SUBWF COUNTER, W выполнит операцию (COUNTER) – (W) и поместит результат в W. По команде SUBWF COUNTER, F выполнится операция (COUNTER) – (W) и результат будет помещён в COUNTER. Оба варианта команды изменяют флаги C, DC, Z. Подробно выполнение команды SUBWF рассмотрено в примерах 1 и 2.

Команда BTFSS f,b выполняется следующим образом: если бит “b” в регистре “f” равен 0, тогда следующая команда выполняется. Если бит b в регистре f равен 1, тогда следующая команда не выполняется и в этом случае команда длится два машинных цикла. Подробно выполнение команды рассмотрено в примере 3.

Пример1. Команда SUBWF COUNTER, W (результат помещается в регистр W).

Случай 1: уменьшаемое (COUNTER) равно вычитаемому (W) результат 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0x00	
COUNTER = 0x02	COUNTER = 0x02	
C любое значение	C = 1;	Результат ноль
Z любое значение	Z = 1	

Случай 2: уменьшаемое (COUNTER) больше вычитаемого (W) результат больше 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0x01	
COUNTER = 0x03	COUNTER = 0x03	
C любое значение	C = 1;	Результат положительный
Z любое значение	Z = 0	

Случай 3: уменьшаемое (COUNTER) меньше вычитаемого (W) результат меньше 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0xFF	
COUNTER = 0x01	COUNTER = 0x01	
C любое значение	C = 0;	Результат отрицательный
Z любое значение	Z = 0	

Пример2. Команда SUBWF COUNTER, F (результат помещается в регистр COUNTER).

Случай 1: уменьшаемое (COUNTER) равно вычитаемому (W) результат 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0x02	
COUNTER = 0x02	COUNTER = 0x00	
C любое значение	C = 1;	Результат ноль
Z любое значение	Z = 1	

Случай 2: уменьшаемое (COUNTER) больше вычитаемого (W) результат больше 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0x02	
COUNTER = 0x03	COUNTER = 0x01	
C любое значение	C = 1;	Результат положительный
Z любое значение	Z = 0	

Случай 3: уменьшаемое (COUNTER) меньше вычитаемого (W) результат меньше 0		
До выполнения команды	После выполнения команды	
W = 0x02	W = 0x02	
COUNTER = 0x01	COUNTER = 0xFF	
C любое значение	C = 0;	Результат отрицательный
Z любое значение	Z = 0	

Пример 3. Команда BTFSS f,b

```
HERE:          BTFSS    STATUS, 2
FALSE:         GOTO     PROCESS_CODE
TRUE:          *****
```

Случай 1: содержимое регистра STATUS = xxxx xx0x (второй бит ноль).		
До выполнения команды	После выполнения команды	
PC = address HERE	PC = address FALSE	
Так как STATUS<2> = 0		

Случай 2: содержимое регистра STATUS = xxxx xx1x (второй бит единица).		
До выполнения команды	После выполнения команды	
PC = address HERE	PC = address TRUE	
Так как STATUS<2> = 1		

Задание

Напишите программу, которая будет сравнивать содержимое двух регистров и выполнять ветвление, в результате которого микроконтроллер будет сигнализировать о равенстве или неравенстве содержимого ячеек памяти. Создайте проект, откомпилируйте программу Project5. Запустите программу в симуляторе MPLAB IDE. Откройте окно симулятора Watch. Промоделируйте работу программы.

Подключите один из семисегментных индикаторов к порту микроконтроллера. Запрограммируйте микроконтроллер и продемонстрируйте работу программы на лабораторном макете.

Порядок выполнения

На диске C:\ в папке Projects создайте папку Project5. В эту папку скопируйте файл Project5 и создайте проект с названием Project5. Откомпилируйте программу. Откройте окно Watch и настройте его для отображения регистров STATUS, PORTD, рабочего регистра W (WREG) и регистра COUNTER (рис. 1). Наведите курсор на регистр STATUS и щёлкните правой кнопкой мыши. В появившемся окне поставьте галочку напротив SFR Bitfield Mouseover (рис. 2). Теперь, если щёлкнуть мышкой на регистре STATUS, будут показываться названия битов этого регистра (рис. 3).

Выберите симулятор MPLAB SIM (рис. 4). Выберите окно настройки симулятора (рис. 5). В появившемся окне выберите вкладку Animation/Realtime Updates (рис. 6) и настройте симулятор. Для этого в графе Animate step time установите значение 1000 мс, поставьте галочку напротив Enable Realtime watch updates и введите цифру 1 в графе x100 msecs.

Установите точку останова на команде btfs (рис. 7). Запустите программу в режиме Animate (рис. 8). Обратите внимание на изменение содержимого регистра COUNTER и рабочего регистра W.

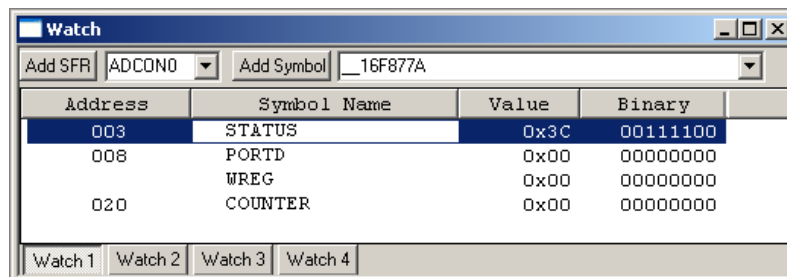


Рис. 1. Вид настроенного окна Watch.

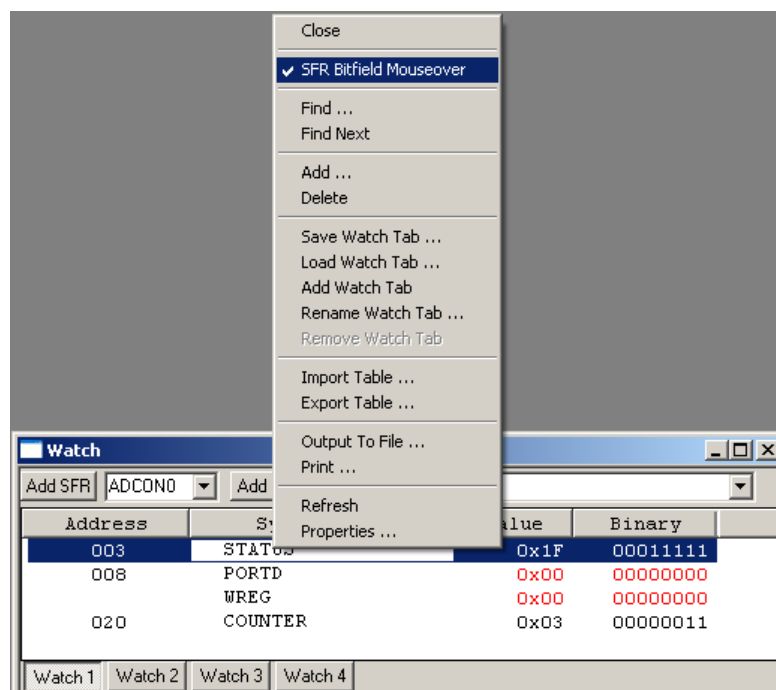


Рис. 2. Вид меню настройки окна Watch.

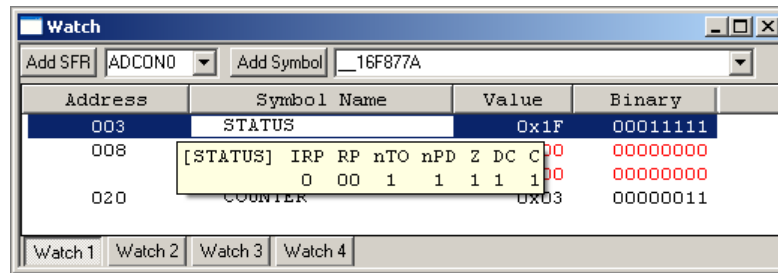


Рис. 3. Вид окна Watch (подпись битов появляется, если навести мышь на регистр).

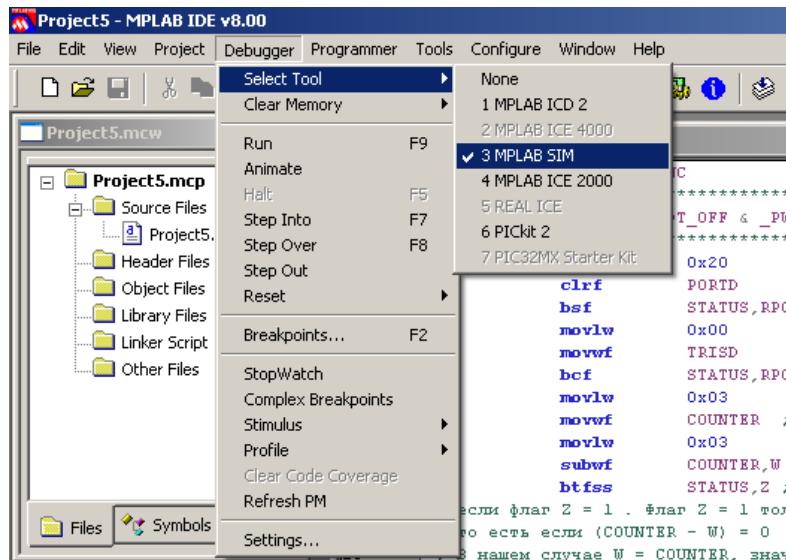


Рис. 4. Выбор симулятора MPLAB SIM.

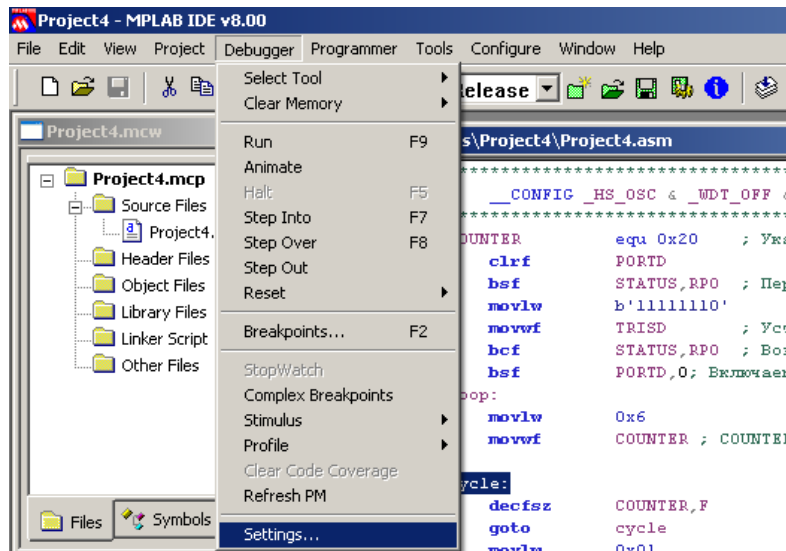


Рис. 5. Выбор окна настройки симулятора.

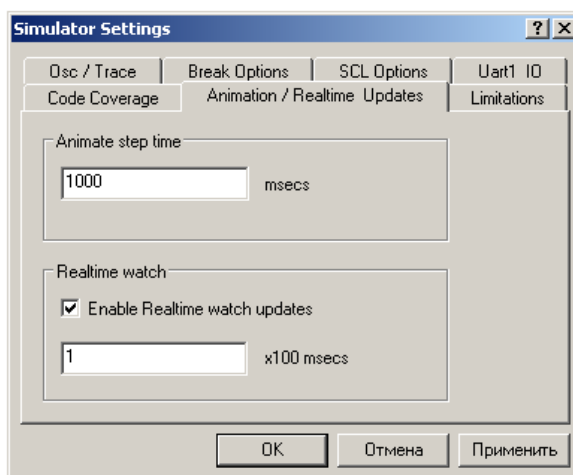


Рис. 6. Настройка симулятора.

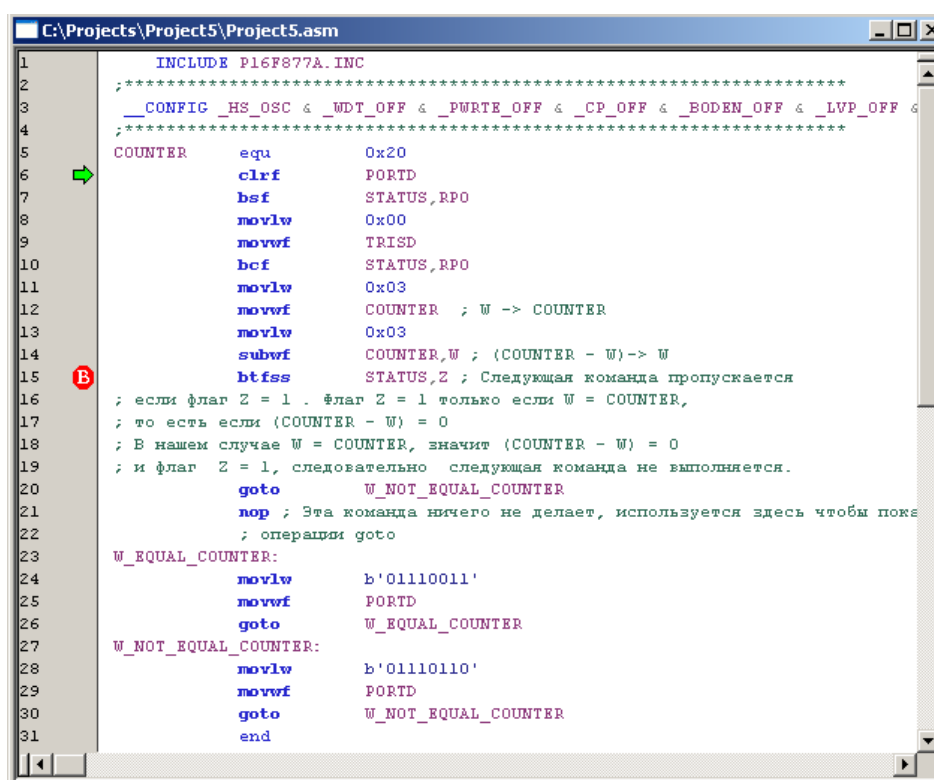


Рис. 7. Вид точки останова (строка 15).

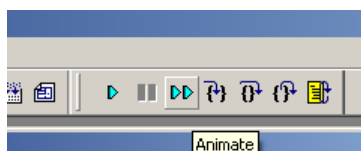


Рис. 8. Кнопка Animate.

После того, как программа дойдёт до точки останова, выполнение программы будет приостановлено. Обратите внимание на флаг Z в регистре STATUS. Так как этот флаг установлен, следующая команда (goto) будет пропущена. Снова запустите программу (рис. 8) – программа перейдёт на метку W_EQUAL_COUNTER. В этой части программы на PORTD выводится двоичное число 01110011, что соответствует букве «P» (равно) на семисегментном индикаторе.

Остановите симулятор кнопкой Halt (рис. 9). Измените программу так, чтобы в регистр COUNTER и в регистр W попадали разные числа. Заново откомпилируйте программу, перезагрузите симулятор кнопкой Reset (рис. 10).

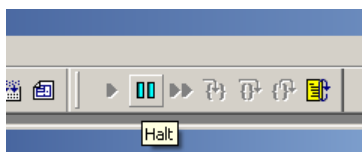


Рис. 9. Кнопка Halt для остановки симулятора.

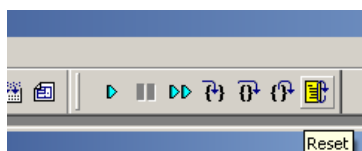


Рис. 10. Кнопка Reset для перезагрузки.

Запустите программу в режиме Animate (рис. 8). Обратите внимание на изменение содержимого регистра COUNTER и рабочего регистра W.

После того, как программа вновь дойдёт до точки останова, выполнение программы будет приостановлено. Обратите внимание, что на этот раз флаг Z в регистре STATUS будет сброшен и следующая команда (goto) будет выполнена. Снова запустите программу (рис. 8) – программа перейдёт на метку W_NOT_EQUAL_COUNTER. В этой части программы на PORTD выводится двоичное число 01110110, что соответствует букве «Н» (не равно) на семисегментном индикаторе.

После симуляции соберите схему рис. 16 на лабораторном макете как показано на рис. 17. После этого в среде разработки выберите имеющийся у вас отладчик, например PICkit 2 (рис. 11).

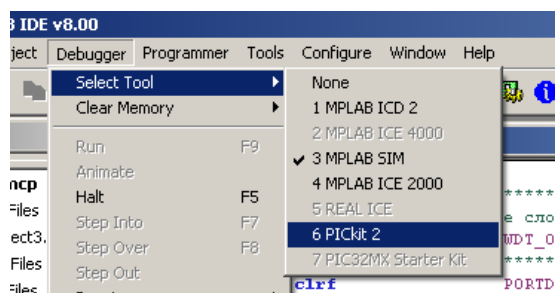


Рис. 11. Выбор отладчика в среде разработки.

Запрограммируйте лабораторный макет (рис. 12). Запустите программу в режиме Animate (рис. 14).

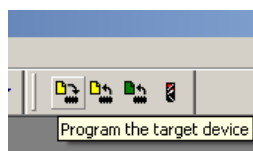


Рис. 12. Кнопка для программирования лабораторного макета.

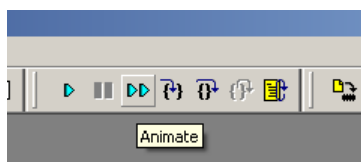


Рис. 14. Кнопка для запуска программы на макете в режиме Animate.

Переделайте программу так, чтобы в регистрах содержалось одинаковое число, запрограммируйте лабораторный макет. Запустите программу на макете и предъявите результат.

Аппаратное обеспечение

В этой работе используется семисегментный индикатор. Семисегментный индикатор состоит из восьми светодиодов и каждый из них работает как обычный светодиод. Только в семисегментном индикаторе светодиоды выполнены в виде полосок. Полоски имеют определённое положение на плоскости индикатора. Комбинируя горящие и не горящие светодиоды можно получать различные символы. Принципиальная электрическая схема семисегментных индикаторов лабораторного макета изображена на рис. 15. Принципиальная электрическая схема для выполнения лабораторной работы изображена на рис. 16. На рис. 17 показаны соединения на плате лабораторного макета.

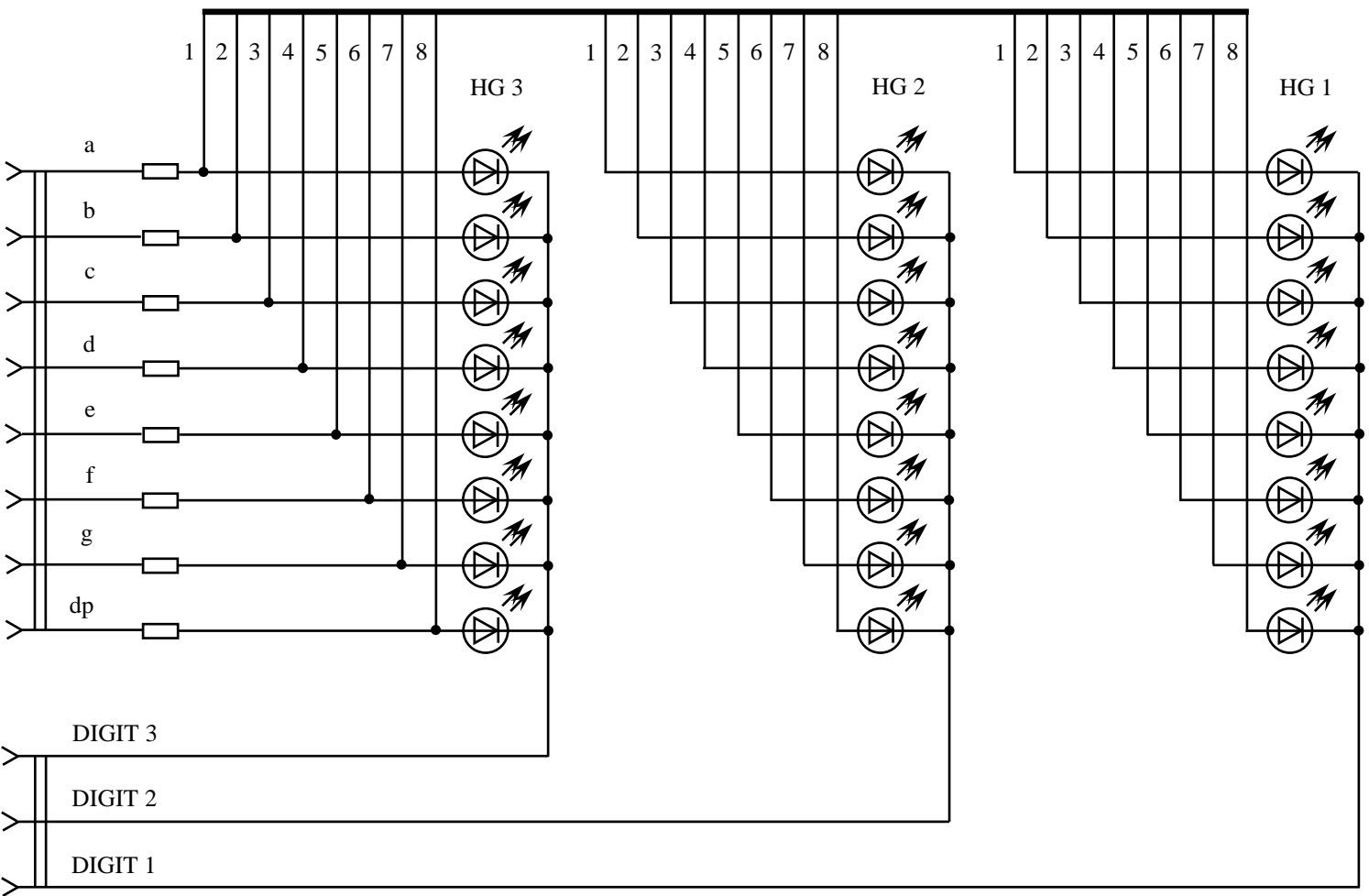


Рис. 15. Принципиальная электрическая схема семисегментных индикаторов.

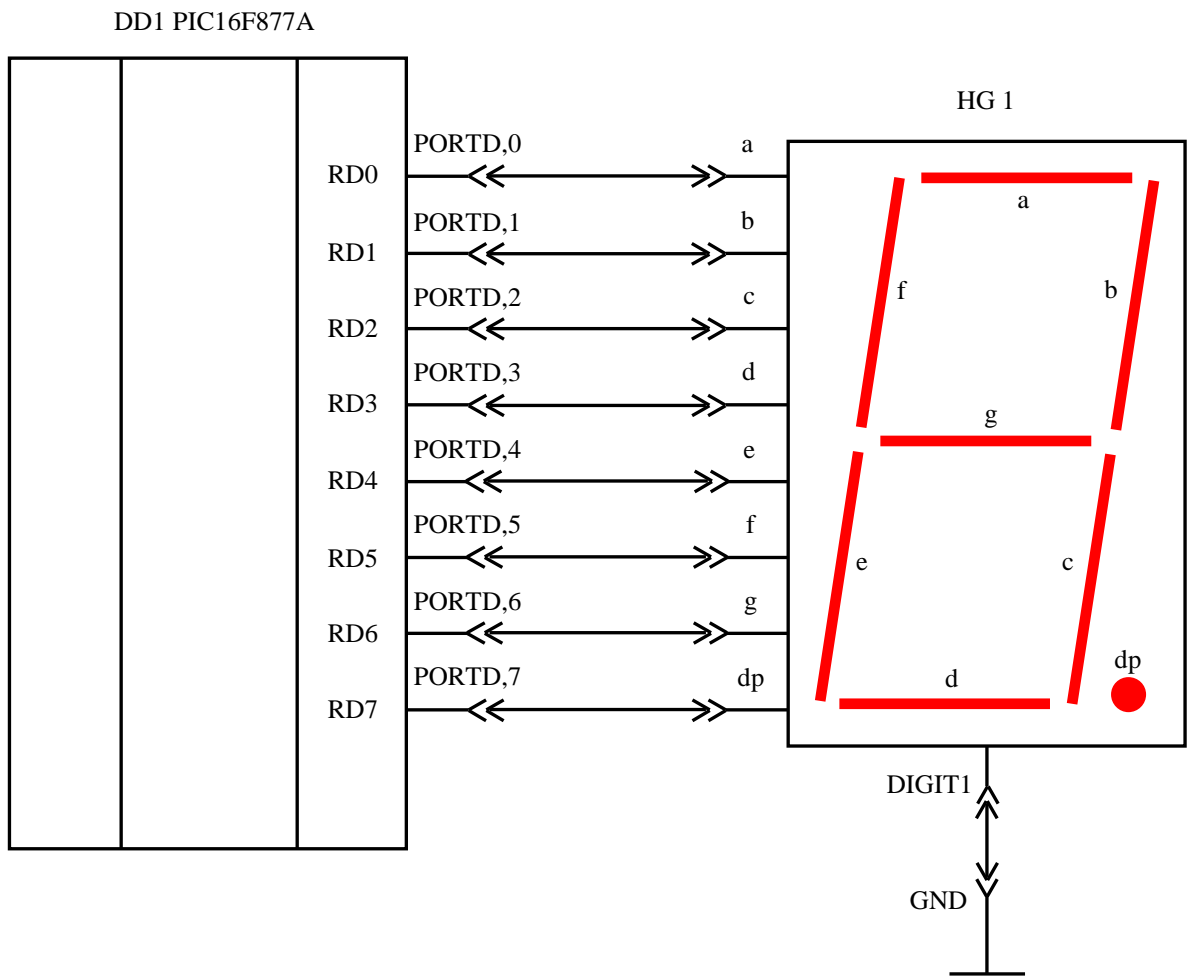


Рис. 16. Принципиальная электрическая схема для выполнения лабораторной работы.

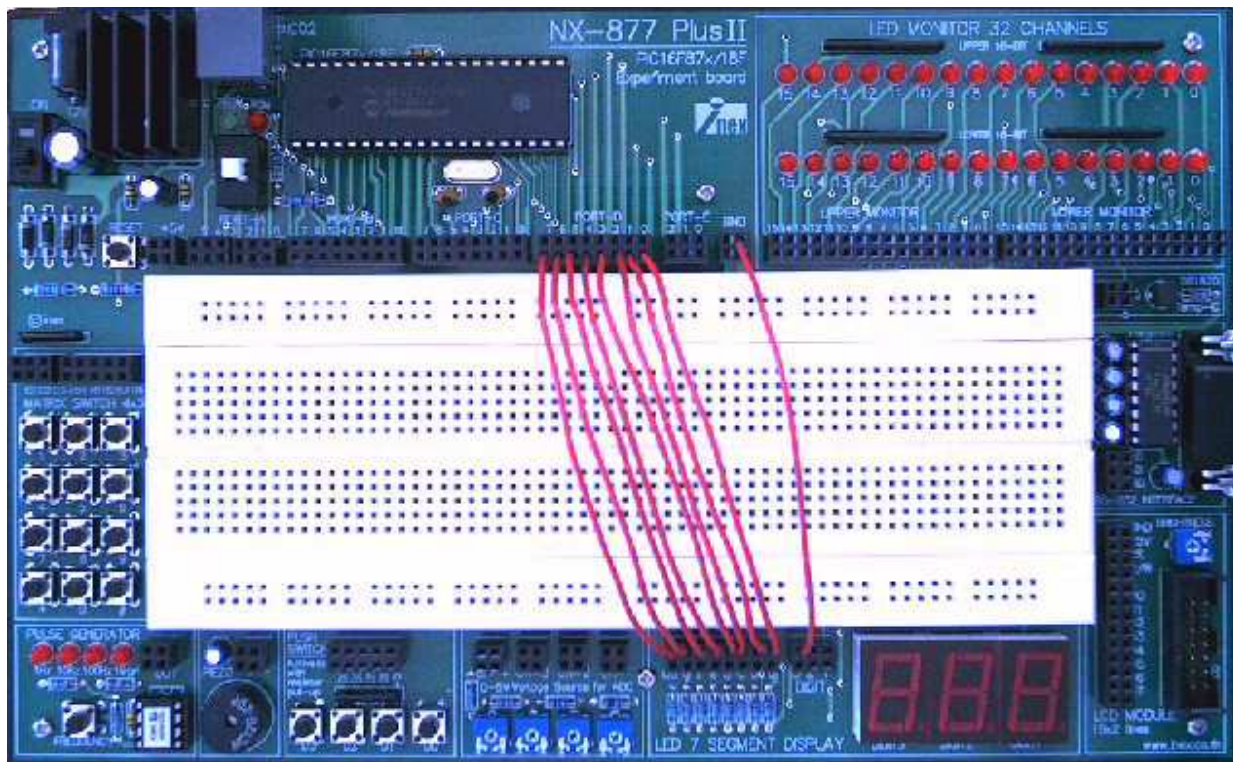


Рис. 17. Схема, собранная на лабораторном макете.

Программное обеспечение

Алгоритм программы изображён на рис. 18.

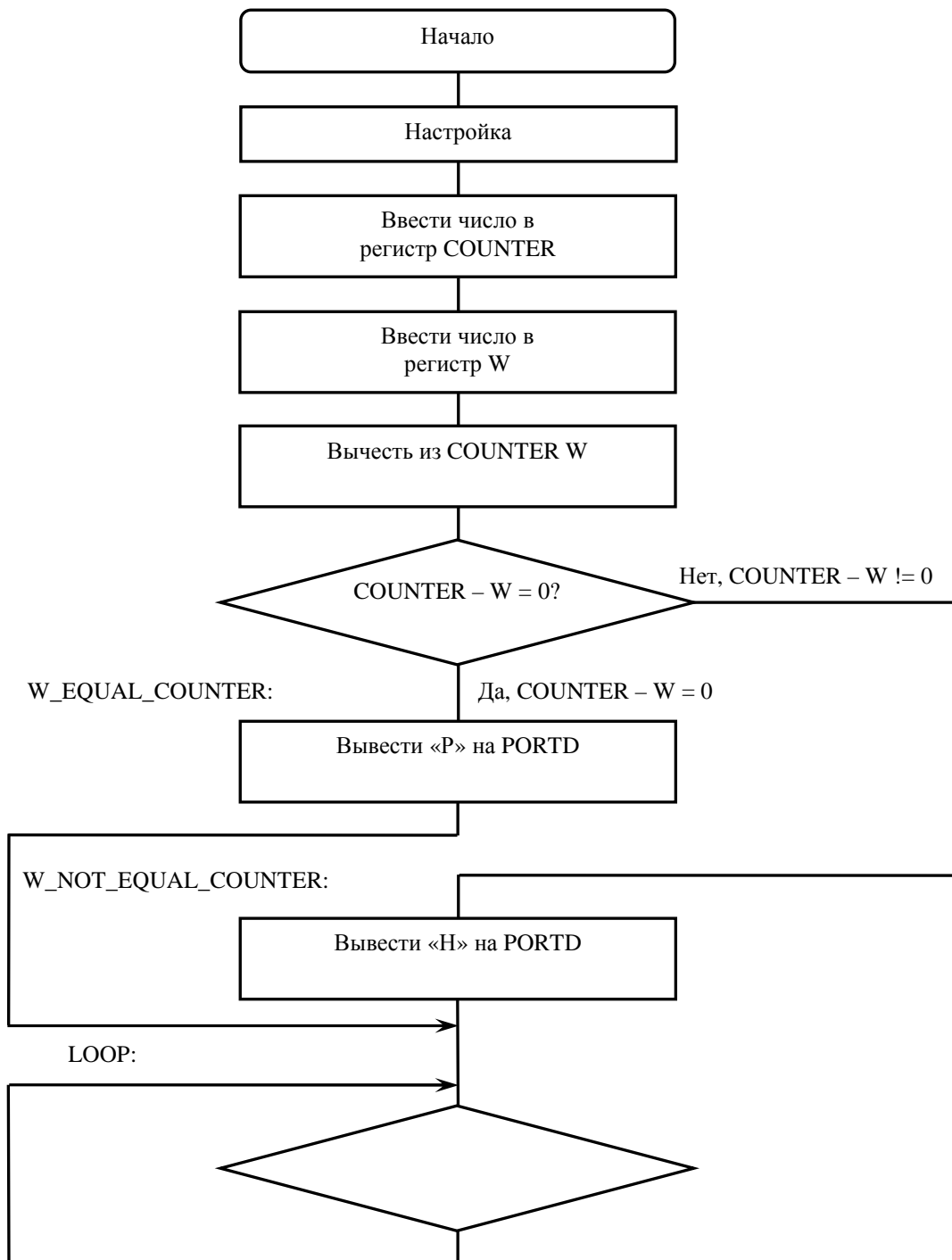


Рис. 18. Алгоритм программы Project5.

Текст файла Project5.ASM

INCLUDE P16F877A.INC

```
*****
;_CONFIG_HS_OSC & _WDT_OFF & _PWRTE_OFF & _CP_OFF & _BODEN_OFF &
_LVP_OFF & _CPD_OFF & _DEBUG_ON
*****
COUNTER      equ      0x20
              clrf     PORTD
              bsf     STATUS,RP0
              movlw   0x00
              movwf   TRISD
              bcf     STATUS,RP0
              movlw   0x03
              movwf   COUNTER ; W -> COUNTER
              movlw   0x03
              subwf   COUNTER,W ; (COUNTER - W)-> W
              btfss  STATUS,Z ; Следующая команда пропускается
; если флаг Z = 1 . Флаг Z = 1 только если W = COUNTER,
; то есть если (COUNTER - W) = 0
; В нашем случае W = COUNTER, значит (COUNTER - W) = 0
; и флаг Z = 1, следовательно следующая команда не выполняется.
              goto    W_NOT_EQUAL_COUNTER
              nop     ; Эта команда ничего не делает, используется здесь
; чтобы показать пропуск команды goto
W_EQUAL_COUNTER:
              movlw   b'01110011'
              movwf   PORTD
              goto    LOOP
W_NOT_EQUAL_COUNTER:
              movlw   b'01110110'
              movwf   PORTD
LOOP:         goto    LOOP
              end
```

Индивидуальные задания

Напишите программу для сравнения содержимого двух ячеек памяти данных. Если содержимое ячейки памяти COUNTER1 больше COUNTER2, то должна выводиться буква «b» (bigger). Если содержимое COUNTER1 меньше COUNTER2 выводится буква «S» (smaller). Если содержимое ячеек равно, тогда должна выводиться буква «E» (equal)

Контрольные вопросы

1. Что является уменьшаемым, а что является вычитаемым в команде SUBWF f,W?
2. При каких условиях устанавливается бит Z в регистре STATUS?
3. При каких условиях устанавливается бит C в регистре STATUS?
4. Каков механизм работы оператора условного перехода (дать описание события, которое происходит, если условие выполнено и что происходит, если условие не выполнено)?
5. Объясните, как работают команды btfss и btfsc?

Оглавление:	
ЛАБОРАТОРНАЯ РАБОТА №3 «УСЛОВНЫЕ ПЕРЕХОДЫ»	3
Цель работы	3
Теоретические основы	3
Задание.....	5
Порядок выполнения.....	6
Аппаратное обеспечение	10
Программное обеспечение.....	13
Индивидуальные задания	14
Контрольные вопросы.....	14
Список рисунков:	
Рис. 1. Вид настроенного окна Watch.....	6
Рис. 2. Вид меню настройки окна Watch.....	6
Рис. 3. Вид окна Watch (подпись битов появляется, если навести мышь на регистр).	7
Рис. 4. Выбор симулятора MPLAB SIM.....	7
Рис. 5. Выбор окна настройки симулятора.	7
Рис. 6. Настройка симулятора.	8
Рис. 7. Вид точки останова (строка 15).	8
Рис. 8. Кнопка Animate.....	8
Рис. 9. Кнопка Halt для остановки симулятора.	9
Рис. 10. Кнопка Reset для перезагрузки.	9
Рис. 11. Выбор отладчика в среде разработки.	9
Рис. 12. Кнопка для программирования лабораторного макета.	9
Рис. 14. Кнопка для запуска программы на макете в режиме Animate.	10
Рис. 15. Принципиальная электрическая схема семисегментных индикаторов.	11
Рис. 16. Принципиальная электрическая схема для выполнения лабораторной работы.	12
Рис. 17. Схема, собранная на лабораторном макете.	12
Рис. 18. Алгоритм программы Project5.	13

