

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра промышленной электроники и
информационно-измерительной техники

А.В. ХЛУДЕНЕВ

ОТЛАДОЧНЫЕ СРЕДСТВА МИКРОПРОЦЕССОРНЫХ СИСТЕМ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Рекомендовано к изданию Редакционно-издательским советом
государственного образовательного учреждения высшего профессионального
образования «Оренбургский государственный университет»

Оренбург 2009

УДК 621.396.6
ББК 32.844-02
Х60

Рецензент

доктор технических наук, профессор В.Н. Булатов

Хлуденев, А.В.
Х60 **Отладочные средства микропроцессорных систем: методические указания/ А.В.Хлуденев. – Оренбург: ИПК ГОУ ОГУ, 2009. - 32 с.**

Методические указания содержат рекомендации по выполнению лабораторных работ по дисциплине «Отладочные средства микропроцессорных систем».

Методические указания предназначены для студентов, обучающихся по программам высшего профессионального образования по специальности 210106 «Промышленная электроника», а также могут быть использованы студентами других специальностей, связанных с использованием и разработкой встроенных средств микропроцессорной техники.

ББК 32.844-02

© Хлуденев А.В., 2009
© ГОУ ВПО ОГУ, 2009

Содержание

1 Отладка простых программ	4
2 Трансляция и отладка программ сложной структуры	8
3 Отладка программы формирования интервалов времени	10
4 Отладка программы дискретного ввода-вывода	14
5 Отладка программы аналогового ввода	17
6 Отладка программы вывода символьной информации	20
7 Отладка программы, использующей прерывания	23
8 Отладка программной реализации микропрограммного автомата	26
Список использованных источников	32

1 Отладка простых программ

1.1 Описание работы с кратким изложением теории

Цель работы:

- освоить работу в среде системы MPLAB IDE;
- приобрести практические навыки отладки простых программ.

Объектом отладки в работе является программа генератора чисел. Числовая последовательность (в соответствии с вариантом задания) формируется на линиях порта PORTD.

MPLAB IDE – это интегрированная среда разработки программ для микроконтроллеров (МК) PICmicro фирмы Microchip, позволяющая писать, компилировать и отлаживать программы [1]. MPLAB IDE включает несколько инструментальных средств:

- *Project Manager* (менеджер проекта) необходим, чтобы создать проект и работать с файлами, связанными с проектом;
- *Editor* (редактор) необходим для создания и редактирования исходных текстов программ, а также используется как окно отладчика;
- *Language Tools* (языковой инструмент) – позволяет работать с кросс-асемблером MPASM или кросс-компиляторами;
- *Debugger* (отладчик) – совместно с редактором реализует механизмы управляемости и наблюдаемости процесса отладки целевых программ;
- *Execution Engines* (механизм исполнения команд) – программные симуляторы, позволяющие выполнять команды МК на персональном компьютере и имитировать работу периферийных модулей.

Подобно другим инструментальным средствам разработки программ MPLAB IDE содержит встроенный отладчик программ. Но в отличие от резидентных систем в процессе отладки используется не процессор компьютера, а МК PICmicro или его программно-логическая модель. В системе MPLAB IDE реализованы следующие режимы работы отладчика:

- *MPLAB-SIM Simulator* – отладка средствами симулятора, который моделирует работу процессорного ядра и периферийных модулей МК PICmicro;
- *MPLAB-ICE Emulator* – отладка средствами внутрисхемного эмулятора (аппаратного блока для замещения МК PICmicro в реальном времени в отлаживаемой системе);
- *MPLAB-ICD Debugger* - отладка средствами внутрисхемного отладчика (аппаратного блока, позволяющего использовать встроенные в МК PICmicro средства отладки в реальном времени в отлаживаемой системе).

Основное окно системы MPLAB IDE выглядит аналогично большинству инструментальных приложений Windows. Вверху расположена панель меню, а внизу строка состояния. В строке состояния отображается текущая настройка и состояние отладочной системы. Для разработки доступны все инструменты, в том числе для отладки исполняемого кода.

1.2 Создание нового проекта и трансляция программы

В процессе разработки программы формируется проект - множество взаимосвязанных файлов, включая исходный текст программы <имя>.asm, загрузочный модуль <имя>.hex и другие файлы. Все эти файлы удобно хранить в отдельной папке. Взаимосвязи файлов содержатся в корневом файле проекта.

В папке PIC создайте свою рабочую папку, а в ней папку lr1. Скопируйте в нее подготовленный файл с исходным текстом программы lr1.asm. Создайте новый проект, для этого выберите команду *Project > Project Wizard* и последовательно выполните действия:

- задайте тип микроконтроллера PIC16F877;
- задайте транслятор Microchip MPASM Toolsuite;
- задайте имя проекта lr1 и папку для нового проекта;
- включите в проект исходный текст программы lr1.asm.

Откройте окно с исходным текстом программы, для этого в окне менеджера проекта lr1.mcw дважды щелкните по ярлыку lr1.asm в папке *Source Files* или выполните команду *File>Open*.

Выполните трансляцию программы (команда *Project>Make*). На закладке *Build* в окне *Output* будет приведена командная строка, посланная ассемблеру, сообщения об ошибках или успешном завершении трансляции.

Если программа содержала синтаксические и другие формальные ошибки, в окне будет выведен список сообщений об ошибках (ERROR). Если дважды щелкнуть мышкой на сообщении об ошибке, то курсор автоматически перейдет к соответствующему оператору в исходном тексте. Если в программе нет подобных ошибок, ассемблер выдаст сообщение BUILD SUCCEEDED. Это означает, что получен загрузочный модуль программы, готовый для выполнения микроконтроллером. Но это не означает, что программа не содержит логических ошибок, и будет работать правильно. Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки.

1.3 Отладка программы

В данной лабораторной работе предусмотрено выполнение только автономной отладки программы на программно-логической модели МК. Активировать инструмент *MPLAB-SIM Simulator* можно командой *Debugger>Select Tool>MPLAB SIM*.

Существуют различные методы тестирования программ, направленные на проверку их правильности и выявление возможных ошибок. Но практически все они основаны на отслеживании выполнения программы и сравнении с ожидаемым ходом выполнения. Для отладки программ обычно применяют два режима:

- пошаговая отладка;
- выполнение программы (прогон) до точки останова.

При пошаговой отладке выполняется один оператор программы, и затем контролируются те переменные, на которые должен воздействовать данный оператор.

Отдельный участок программы можно выполнить в режиме прогона. Точку останова устанавливают на операторе, при выполнении которого необходимо проверить значения переменных или просто проконтролировать, передается ли управление данному оператору.

Для отслеживания содержимого ограниченного набора ячеек памяти удобно использовать смотровое окно. Чтобы создать смотровое окно выполните команду *View> Watch*. В верхней части окна *Watch* находятся прокручиваемые списки регистров специальных функций (SFR) и идентификаторов, используемых в программе (Symbol). Из левого списка выберите ячейку PORTD, добавьте в окно наблюдения нажатием кнопки *Add SFR*. Из правого списка выберите ячейку Cnt, добавьте в окно наблюдения нажатием кнопки *Add Symbol*.

Инициализируйте МК нажатием кнопки *Reset*. Программный счетчик будет сброшен в ноль. Средствами символьного отладчика строка исходного кода, соответствующая этому адресу, отмечена указателем (зеленой стрелкой). В строке состояния MPLAB IDE отображаются:

- режим работы отладчика (MPLAB SIM);
- тип МК (PIC16F877);
- содержимое программного счетчика PC;
- содержимое аккумулятора W;
- состояние флагов Z dc с (заглавной буквой – установлен, строчной – сброшен);
- тактовая частота МК (20 MHz);
- активный банк памяти данных (bank 0).

Нажмите кнопку *Step Into*. Будет выполнена первая команда программы. Программный счетчик будет показывать PC:0x01, указатель переместится на следующую исполняемую команду. Для пошагового выполнения можно также использовать функциональную клавишу <F7>. Нажимайте клавишу <F7> несколько раз и наблюдайте, как перемещается по тексту указатель, изменяются показания программного счетчика и значения переменных в смотровом окне.

Для регистрации значений переменных в процессе прогона программы можно использовать память трассы. Установите курсор на строке с оператором *goto Loop*. Нажмите правую кнопку мыши и в появившемся меню выберите опцию *Set Breakpoint* (точка останова). Слева от этой строки появится символ “B” красного цвета, сигнализируя, что сразу перед исполнением команды в этой строке выполнение программы остановится.

Продолжите выполнение программы в режиме прогона командой *Run* или клавишей <F9>. При достижении точки останова симулятор остановится на выделенном операторе (но эта команда не выполнится).

Открыть окно памяти трассы можно командой *View> Simulator Trace*. Каждая запись памяти трассы содержит:

- адрес команды (Addr);

- код команды (Op);
- метку (Label);
- дизассемблерное представление команды (Instruction);
- адрес и содержимое ячейки источника (SA и SD);
- адрес и содержимое ячейки приемника (DA и DD);
- текущее время в циклах (Cycles).

Содержимое буфера трассировки можно сохранить в файле командой *Output to File*, очистить буфер трассировки можно командой *Clear Trace Buffer* (панель команд открывается правой кнопкой мыши).

Если нажать клавишу <F9> еще раз, симуляция программы продолжится до следующего достижения точки останова. Разумеется, когда программа выполняется и не достигла точки останова, ее можно остановить командой *Halt* или нажатием клавиши <F5>.

По содержимому окна *Watch* и буфера трассировки определите корректность выполнения программы. При выявлении несоответствия хода выполнения программы ожидаемому, необходимо локализовать ошибку в программе. Для этого необходимо:

- определить шаг (и соответствующий оператор программы), начиная с которого выполнение программы отклоняется от ожидаемого;
- проанализировать причину этого отклонения;
- исправить ошибку в исходном тексте;
- выполнить трансляцию программы.

При последующем тестировании необходимо убедиться в корректности выполнения программы. После выполнения работы закройте проект командой *Project>Close*.

1.4 Содержание отчета

Отчет должен содержать:

- схему алгоритма программы;
- исходный текст программы;
- трассу программы;
- выводы о корректности программы.

1.5 Контрольные вопросы

1.5.1 Перечислите основные инструментальные средства и режимы работы отладчика системы MPLAB IDE.

1.5.2 Перечислите основные режимы отладки программ в среде MPLAB IDE?

1.5.3 Поясните по схеме алгоритма работу программы.

1.5.4 Поясните содержание трассы программы?

2 Трансляция и отладка программ сложной структуры

2.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения трансляции и автономной отладки в среде MPLAB IDE программ сложной структуры.

Содержанием работы является трансляция и автономная отладка программы сложной структуры для МК типа PIC16F877. Программа имеет сложную структуру и состоит из основной программы и вызываемой подпрограммы умножения восьмиразрядных чисел.

Исследуются два варианта трансляции исходных текстов одной и той же программы сложной структуры [2]:

- для прямой генерации абсолютного кода, который может быть загружен непосредственно в память программ микроконтроллера;
- для генерации объектных файлов, которые необходимо связывать при помощи редактора связей MPLINK.

2.2 Описание последовательности выполнения

2.2.1 Трансляция и отладка программы из одного исходного файла

В своей рабочей папке создайте папку lr2_1. Скопируйте в нее подготовленный файл с исходным текстом программы lr2_1.asm. Создайте новый проект с помощью мастера *Project > Project Wizard*. Задайте имя проекта lr2_1, включите в проект исходный текст программы lr2_1.asm.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

В результате трансляции формируются загрузочный модуль программы, листинг трансляции lr2_1.lst. Проанализируйте листинг программы, определите объем полученного кода и число используемых ячеек памяти данных. Включите листинг в отчет.

Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки. Активируйте инструмент *MPLAB-SIM Simulator* командой *Debugger> Select Tool>MPLAB SIM*. Создайте смотровое окно командой *View> Watch*. Включите в смотровое окно основные ячейки:

- mulcnd;
- mulplr;
- H_byte;
- L_byte;
- count;
- H_sum;
- L_sum.

Откройте окно стека командой *View>Hardware Stack*. Выполните проверку работы программы в пошаговом режиме или режиме прогона с остановом. В последнем случае определите операторы для точек останова. При отладке программы проследите за изменениями ячеек памяти данных, передачей управления и содержимым стека. Если ход и результаты выполнения программы отличаются от ожидаемых, необходимо локализовать и устранить ошибки.

Закройте проект командой *Project>Close*.

2.2.2 Трансляция и отладка программы из нескольких исходными файлов

В своей рабочей папке создайте папку *lr2_2*. Скопируйте в нее подготовленный файл с исходным текстом программы *lr2_2.asm* и файл подпрограммы умножения *mul8x8.asm*. Создайте новый проект с помощью мастера *Project > Project Wizard*. Задайте имя проекта *lr2_2*, включите в проект файлы исходных текстов программы *lr2_2.asm* и подпрограммы *mul8x8.asm*.

Откройте окна с исходными текстами программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

В результате трансляции формируются объектные модули и листинги трансляции *lr2_2.lst* и *mul8x8.lst*. В результате работы линкера формируется загрузочный модуль и карта памяти *lr2_2.map*. Проанализируйте листинги и карту памяти программы, определите объем полученного кода и число используемых ячеек памяти данных. Включите листинги и карту памяти в отчет.

Выполните проверку работы программы, поиск и исправление возможных ошибок. Закройте проект командой *Project>Close*.

2.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма;
- листинги трансляции, карту памяти;
- результаты проверки работы программы, выводы.

2.4 Контрольные вопросы

2.4.1 Поясните особенности выполнения программ сложной структуры.

2.4.2 Как может выполняться в среде MPLAB IDE трансляция программ сложной структуры?

2.4.3 Перечислите основные отличия правил формирования исходных текстов программ на языке ассемблера при трансляции в абсолютный или объектный код?

2.4.4 Поясните назначение стека.

2.4.5 Поясните по схеме алгоритма работу программы.

3 Отладка программы формирования интервалов времени

3.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки программ, использующих таймер.

Содержанием работы является автономная и комплексная отладка программы для МК типа PIC16F877 [3] с использованием интегрированной системы MPLAB IDE и внутрисхемного отладчика MICD2. Объектом отладки в работе является программа формирования интервалов времени, через которую происходит переключение выходной линии RD0 порта PORTD (в соответствии с вариантом задания). Для отсчета интервалов времени используется встроенный таймер TMR0.

При выполнении автономной отладки программу выполняет симулятор, взаимодействующий с отладчиком системы MPLAB IDE. При выполнении комплексной отладки программу выполняет реальный МК, который может быть установлен в различных устройствах. На лабораторном занятии комплексная отладка программы выполняется на отладочном стенде. Для обеспечения взаимодействия МК с отладчиком системы MPLAB IDE необходимы специальные аппаратные средства поддержки, в качестве которых используется внутрисхемный отладчик MPLAB ICD2 [4] или его аналог MICD2.

Схема подключения модуля MICD2 к компьютеру и отлаживаемому микропроцессорному устройству (МПУ) - отладочному стенду приведена на рисунке 3.1. Питание отлаживаемого устройства будет осуществляться от модуля MICD2, который, в свою очередь, получает питание от блока питания (АС-DC адаптера 9 В). Отлаживаемое устройство и модуль MICD2 имеют открытые токоведущие части, находящиеся под напряжением не более 12 В. Важно помнить, что при коммутации (подключении или отключении) устройств на линиях связи могут кратковременно формироваться броски напряжения, многократно превышающие номинальные значения уровней напряжения. Поэтому при любых коммутациях питание модуля MICD2 должно быть отключено.

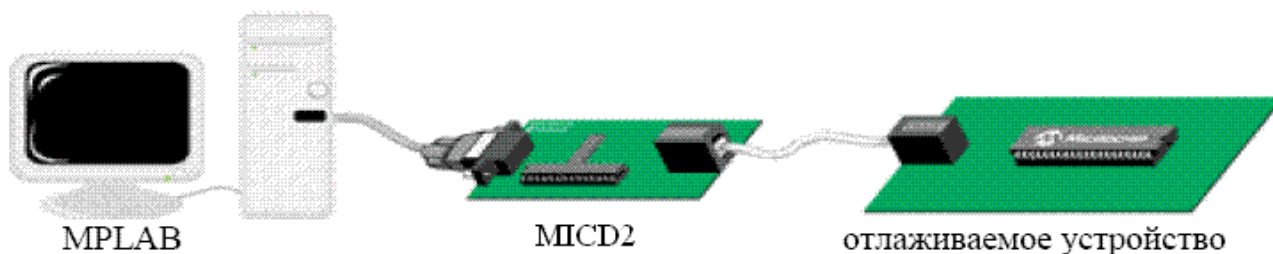


Рисунок 3.1

Директива `__CONFIG` в программе позволяет задать необходимые значения разрядов слова конфигурации PIC16F877 (адрес H'2007'). После трансляции их значения можно наблюдать и редактировать в окне `Configure>`

Configuration Bits. Требуемые значения разрядов для обеспечения возможности отладки с помощью отладчика MICD2 приведены в таблице 3.1.

Таблица 3.1 - Разряды слова конфигурации

Пункт	Опция
Oscillator	Кварцевый резонатор (XT)
Watchdog Timer	Сторожевой таймер (WDT) должен быть выключен (Off)
Power Up Timer	Таймер включения (PWRT) должен быть включен (On).
Brown Out Detect	Детектор дрожания напряжения питания (BOD) должен быть выключен (Off)
Low Voltage Program	Низковольтное программирование должно быть запрещено (Disabled)
Flash Memory Write	Запись Flash Memory должна быть разрешена (Enabled)
Data EE Read Protect	Защита данных EEPROM должна быть выключена (Off)
Code Protect	Защита кода программы должна быть выключена (Off)

3.2 Описание последовательности выполнения

3.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы *lr3.asm*. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта *lr3*, включите в проект исходный текст программы *lr3.asm*.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

3.2.2 Активируйте отладчик MPLAB IDE для работы с симулятором (команда *Debugger>Select Tool>MPLAB SIM*). Настройте симулятор для выполнения автономной отладки данного приложения (команда *Debugger>Setting*). В окне *Simulator Setting* на закладке *Osc/Trace* задайте значение тактовой частоты *Processor Frequency* 3.6864 MHz.

Создайте окно наблюдения интересующих ячеек памяти данных (команда *View>Watch*). В левом окне выберите PORTD и нажмите кнопку *Add SFR*. В правом окне выберите Count05s и нажмите кнопку *Add Symbol*.

Контролировать текущие значения ячеек можно в окне наблюдения. Выполните команду *View>Watch*, чтобы создать окно наблюдения интересующих ячеек памяти данных: PORTD, Count05s.

Контролировать текущее время выполнения программы, а также время выполнения отдельных ее участков, измерять интервалы времени между различными событиями можно с помощью секундомера. Откройте окно секундомера *Debugger>Stopwatch*. В окне секундомера можно увидеть текущий номер

цикла, текущее время выполнения программы и тактовую частоту микроконтроллера.

Установите точки останова на оператора вызова подпрограммы формирования временного интервала (`call Dly`) и возврата из подпрограммы (`return`).

Выполните сброс МК (`Debugger>Reset>Processor Reset`), а затем активируйте выполнение программы в режиме прогона (`Debugger>Run`). При останове программы в первой точке сбросьте показания секундомера в окне *StopWatch* кнопкой *Zero*. Активируйте дальнейшее выполнение программы в режиме прогона (`Debugger>Run`). При останове программы по адресу команды `return` по показаниям секундомера контролируйте время выполнения подпрограммы `Dly`. При отклонении измеренного значения от расчетного проверьте правильность настройки таймера `TMR0` и выполните отладку подпрограммы `Dly`. Измеренное значение временной задержки занесите в отчет.

При отладке программы в среде симулятора следует учитывать, что скорость моделирования зависит от производительности компьютера.

3.2.3 Подключите отладчик `MICD2` к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику `MICD2` источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика `MICD2`!

Активируйте отладчик `MPLAB IDE` для работы с `MICD2` (команда `Debugger>Select Tool>MPLAB ICD2`). Настроить отладчик `MPLAB IDE` для выполнения комплексной отладки данного приложения можно с помощью инструмента *MPLAB ICD2 Setup Wizard*. Для связи с отладчиком `MICD2` выберите порт `Com1` и скорость обмена *Baud Rate 57600*. Далее задайте способ подключения источника питания через отладчик (*Power target from the MPLAB ICD2*). Автоматическое установление связи с отладчиком `MICD2` необходимо запретить (снять флажок на опции `MPLAB IDE automatically connects to MPLAB ICD2`). Автоматическую загрузку программного обеспечения отладчика необходимо разрешить (установить флажок на опции `MPLAB IDE automatically downloads the required operating system`). Далее необходимо принять во внимание сообщение, что для работы с отладчиком `MPLAB ICD2` буферы `FIFO Com` порта должны быть отключены.

Установите связь с отладчиком `MICD2` командой `Debugger>Connect`. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (*Operating System*) и самотестирования можно приступить к загрузке приложения в целевой микроконтроллер отладочного стенда по команде `Debugger>Program`. При успешном завершении программирования МК в окне *Output* на закладке *MPLAB ICD2* появится сообщение:

```
Connecting to debug executive
...Programming succeeded
дата, время
MPLAB ICD 2 Ready.
```

MPLAB-ICD2 позволяет МК PIC16F877 выполнять программу в режиме прогона (реального времени), пошаговом режиме или анимации. Начните с режима выполнения в реальном времени. Нажмите кнопку панели *Run* или выберите команду *Debugger>Run*. С помощью часов оцените длительность интервалов времени между моментами переключения светодиода, подключенного к линии 0 порта PORTD. Сравните с расчетными значениями. Нажмите кнопку панели *Halt* или выберите команду *Debug>Halt*, чтобы остановить выполнение программы.

Установите точку останова на команде *call Dly*. Запускайте программу на выполнение в режиме прогона до полного выполнения одного цикла основной программы. При достижении точки останова регистрируйте в отчете значение параметра, передаваемого подпрограмме *Dly* - содержимое аккумулятора.

Отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от СОМ-порта компьютера.** Закройте проект командой *Project>Close*.

3.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма программы;
- исходный текст программы;
- результаты отслеживания выполнения программы при автономной и комплексной отладке;
- расчетные и измеренные значения задержек;
- выводы.

3.4 Контрольные вопросы

3.4.1 Поясните, каким образом можно настроить таймер TMR0 на требуемый режим работы.

3.4.2 Поясните, каким образом выполняется расчет задержек.

3.4.3 Поясните по схеме алгоритма работу программы.

3.4.4 Поясните назначение слова конфигурации МК, его отдельных разрядов. Каким образом можно задать требуемые значения разрядов?

3.4.5 Перечислите основные режимы работы отладчика системы MPLAB IDE. Как выполняется настройка среды MPLAB IDE на требуемый режим отладки?

3.4.6 Каким образом можно оценить длительность формируемых временных интервалов с использованием симулятора?

3.4.7 Каким образом можно отслеживать выполнение программы с использованием внутрисхемного отладчика MICD2?

4 Отладка программы дискретного ввода-вывода

4.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки программ дискретного ввода-вывода в среде MPLAB IDE.

Содержанием работы является отладка программы для МК типа PIC16F877, которая анализирует состояние кнопок, подключенных к линиям порта RB1 и RB2. Нажатие с последующим отпусканием кнопок RB1 и RB2 вызывает разнонаправленную модификацию содержимого ячейки Cnt (в соответствии с вариантом задания), которое отображается на светоизлучающих диодах, подключенных к линиям порта PORTD. Активному (нажатому) состоянию кнопки соответствует низкий потенциал на линии порта PORTB. Пассивному (отжатому) состоянию кнопки соответствует высокий потенциал, который обеспечивается встроенными подтягивающими резисторами.

4.2 Описание последовательности выполнения

4.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы lr4.asm. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта lr4, включите в проект исходный текст программы lr4.asm.

Откройте окно с исходным текстом программы. При выполнении автономной отладки целесообразно отключить функции подавления дребезга контактов кнопок. Для этого достаточно закомментировать операторы вызова подпрограммы задержки D15ms. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

4.2.2 Активируйте отладчик MPLAB IDE для работы с симулятором (команда *Debugger>Select Tool>MPLAB SIM*). Настройте симулятор для выполнения автономной отладки данного приложения (команда *Debugger>Setting*). В окне *Simulator Setting* на закладке *Osc/Trace* задайте значение тактовой частоты *Processor Frequency* 3.6864 MHz.

Выполните команду *View> Watch*, чтобы создать окно наблюдения интересующих ячеек памяти данных: PORTB, PORTD, Cnt.

Для автономной отладки программы необходимо иметь средства имитации нажатия кнопок. Для этих целей можно использовать асинхронные стимулы. Чтобы создать асинхронный стимул, выберите пункт меню *Debugger>Stimulus>New Workbook* для открытия диалогового окна настройки стимулов. В открывшемся окне *Stimulus* на закладке *Asynch* содержится таблица, каждая строка которой позволяет задать атрибуты одного стимула. Установите курсор на первой строке в столбце *Pin/SFR* и задайте линию порта ввода RB1. На второй строке в столбце *Pin/SFR* и задайте линию порта ввода RB2. В столбце *Action* выберите значение *Set High* (высокий уровень) для обоих стимулов. Эти

значения будут имитировать исходное воздействие подтягивающих резисторов. В двух следующих строках таблицы создайте стимулы для RB1 и RB2 для имитации воздействия кнопок. Для этого в столбце *Action* выберите значение *Pulse Low* (импульс низкого потенциала), а в столбце *Width* значение длительности импульса в единицах времени, указанных в столбце *Units*. Длительность этих импульсов в реальном устройстве определяется временем нажатия кнопки и составляет доли секунды. При моделировании целесообразно задать существенно меньшие значения, чтобы сократить время моделирования. Вполне достаточно установить 100 сус.

4.2.3 Необходимо выполнить тестирование программы. Сначала следует задать исходные (пассивные) значения разрядов RB1=1 и RB2=1. Произвольно изменить содержимое PORTB не удастся, поэтому произведите сброс МК (*Debugger*> *Reset*), затем нажмите кнопку *Fire* стимула RB1 и кнопку *Fire* стимула RB2 типа *Set High*, выполните одну команду программы (*Debugger*>*Step Into*). Проконтролируйте установку значений разрядов 1 и 2 порта PORTB в окне наблюдения.

Проверку работы программы удобно выполнять в режиме анимации (быстрого пошагового режима), который активируется командой *Debugger* >*Animate*. В этом режиме после выполнения очередной команды обновляется содержимое всех открытых окон системы MPLAB IDE и только затем управление передается следующей команде программы.

Нажмите кнопку *Fire* стимула RB2 типа *Pulse Low* на панели контроллера стимулов. Проследите за изменением соответствующего разряда PORTB в окне наблюдения. Убедитесь, что «нажатие» кнопки вызывает необходимую модификацию переменной Cnt. Повторите эту проверку необходимое число раз. Выполните аналогичные проверки для стимула RB1 типа *Pulse Low*. Занесите в отчет формируемую последовательность значений переменной Cnt. Остановите процесс выполнения программы нажатием кнопки *Halt*.

4.2.4 Восстановите операторы вызова подпрограммы задержки D15ms в исходном тексте программы. Выполните трансляцию программы. Подключите отладчик MICD2 к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику MICD2 источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика MICD2!

Активируйте отладчик MPLAB IDE для работы с MICD2 (команда *Debugger*>*Select Tool*>*MPLAB ICD2*). С помощью инструмента *MPLAB ICD2 Setup Wizard* настройте отладчик MPLAB IDE для выполнения комплексной отладки данного приложения. Установите связь с отладчиком MICD2 командой *Debugger*>*Connect*. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (Operating System) и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger*>*Program*.

Нажмите кнопку панели *Run* или выберите команду *Debugger>Run*. По состоянию линейки светодиодов, подключенных к выводам PORTD, проверьте реакцию МК на нажатие кнопок «+1» и «-1». Сравните с заданным поведением. Нажмите кнопку панели *Halt* или выберите команду *Debugger>Halt*, чтобы остановить выполнение программы. Сбросьте МК, выбрав *Debugger> Reset> Processor Reset*.

Если результаты проверки Вас разочаровали, то в данном случае, вероятнее всего, причина в настройке портов или дребезге контактов. Проверьте правильность управляющих слов, загруженных в регистры TRISB, TRISD, OPTION_REG. Проверьте правильность реализации функции подавления дребезге контактов. После исправления ошибок и трансляции программы, повторите загрузку целевого приложения в МК и тестирование программы.

В случае успеха отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от Com-порта компьютера.** Закройте проект командой *Project>Close*. В противном случае необходимо локализовать и устранить ошибки в программе.

4.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма программы;
- исходный текст программы;
- граф состояний реализованного МПУ;
- выводы.

4.4 Контрольные вопросы

4.4.1 Поясните по схеме алгоритма работу программы.

4.4.2 Как выполняется автономная отладка программ, выполняющих опрос кнопок, с использованием симулятора?

4.4.3 Как в среде симулятора можно имитировать изменения внешних сигналов на линиях портов ввода МК?

4.4.4 Каким образом выполняется программное подавление «дребезга контактов»?

5 Отладка программы аналогового ввода

5.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки в среде MPLAB IDE программы аналогового ввода.

Объектом отладки в работе является программа, которая выполняет запуск модуля АЦП, обрабатывает результаты преобразования и выводит их в PORTD на индикаторы.

5.2 Описание последовательности выполнения

5.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы `lr5.asm`. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта `lr5`, включите в проект исходный текст программы `lr5.asm`.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

5.2.2 Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки. Активируйте инструмент *MPLAB-SIM Simulator* командой *Debugger>Select Tool>MPLAB SIM*. Настройте симулятор для выполнения автономной отладки данного приложения (команда *Debugger>Setting*). В окне *Simulator Setting* на закладке *Osc/Trace* задайте значение тактовой частоты *Processor Frequency* 3.6864 MHz.

5.2.3 Сформируйте удобную среду для отладки программы. Для автономной отладки программы необходимо иметь средства имитации аналого-цифрового преобразования. Для этих целей можно использовать стимулы регистра типа *Register Injection*. Текстовый файл стимула регистра состоит из колонки значений, которые будут поочередно передаваться паре регистров ADRESH:ADRESL с учетом заданного выравнивания, когда аналого-цифровое преобразование будет выполнено.

Создайте новый текстовый файл - *File > New File*. В безымянном окне в одну колонку введите список из восьми 16-ричных значений результатов 10-разрядного аналого-цифрового преобразования, которые Вы хотите назначить регистрам ADRESH:ADRESL. Подберите эти значения, учитывая возможность контроля обработки результатов аналого-цифрового преобразования программой. Сохраните созданный файл - *File > Save As...* Выберите директорию, в которой хотите сохранить файл, введите имя файла с расширением `txt`.

Выберите пункт меню *Debugger>Stimulus>New Workbook* для открытия диалогового окна настройки стимула регистра. В открывшемся окне выберите закладку *Register Injection*. В указанных столбцах задайте значения:

- *Reg/Var* - ADRESL;
- *Trigger* – Demand;

- *Data Filename* – имя файла стимула;
- *Wrap* – Yes;
- *Format* - Hex.

Нажмите кнопку *Apply*, затем *Save* – сохраните стимул регистра.

Контролировать текущие значения сигналов можно в окне наблюдения. Выполните команду *View> Watch*, чтобы создать окно наблюдения интересных ячеек памяти данных: ADRESH, ADRESL, PORTD.

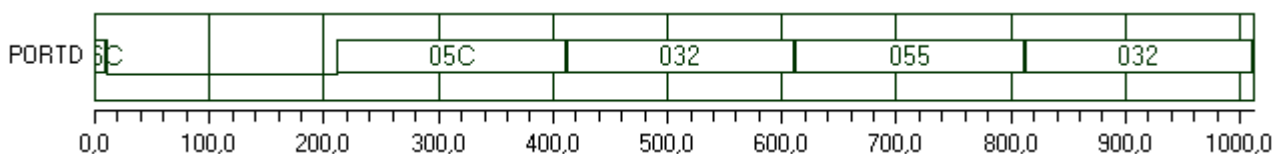
Чтобы получить информацию об изменениях сигналов на выводах МК можно использовать логический анализатор отладчика системы MPLAB IDE. Запуск логического анализатора выполняется по условию совпадения программного счетчика с заданным значением (адресом).

5.2.4 Выполните тестирование программы, используя подготовленные стимулы. В соответствии с планом тестирования при выполнении программы должно произойти 8 запусков модуля АЦП. Это значит, что необходимо выполнить 4 полных цикла программы. Чтобы наблюдать в окне логического анализатора последний результат, необходимо выполнить дополнительный цикл.

Наиболее просто это обеспечить, используя сложные условия останова отладчика системы MPLAB IDE. Выполните команду *Debugger>Complex Breakpoints*. В открывшемся окне *Simulator Complex Breakpoints* нажмите кнопку *Add Breakpoint*. В окне *Set Breakpoint* на закладке *Program Memory* необходимо в окне *Address* задать метку точки останова – вход в подпрограмму обработки прерывания. В окне *Pass Count* выберите условие *Event must occur Count times*, в окне *Count* задайте число событий до останова программы.

Откройте окно логического анализатора командой *View>Logic Analyzer*, нажмите кнопку *Channals* и в открывшемся окне *Configure Channals* нажмите кнопку *Configure Bus(s)*. В открывшемся окне *Configure Bus* нажмите кнопку *New Bus*, в окне *Bus Name* задайте имя группы (шины) – PORTD, выберите сигналы на линиях RD0 - RD7, подтвердите выбор нажатием кнопки ОК. В окне *Configure Channals* выберите группу PORTD, подтвердите выбор нажатием кнопки ОК. Установить переключатель *Trigger Position* на *Start*.

Запустите выполнение программы в автоматическом режиме. После останова программы, используя окно логического анализатора, исследуйте временные диаграммы зарегистрированных сигналов. Убедитесь, что смена значений выходных сигналов на линиях PORTD происходит в соответствии с планом тестирования. Зарегистрированные временные диаграммы для рассматриваемого примера приведены на рисунке 5.1.



При оценке правильности полученных результатов следует учитывать, что выравнивание результата аналого-цифрового преобразования и отбрасыва-

ние двух младших разрядов эквивалентно делению заданного в файле стимула значения на 4.

Установите точки останова на операторах выбора аналоговых каналов и операторах запуска преобразования. Используя секундомер *Debugger> Stopwatch*, оцените интервалы времени, необходимые для нормальной работы модуля АЦП, а также период дискретизации аналоговых сигналов. В случае необходимости скорректируйте длительности задержек.

5.2.5 Подключите отладчик MICD2 к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику MICD2 источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика MICD2!

Активируйте отладчик MPLAB IDE для работы с MICD2 (команда *Debugger>Select Tool>MPLAB ICD2*). С помощью инструмента *MPLAB ICD2 Setup Wizard* настройте отладчик MPLAB IDE для выполнения комплексной отладки данного приложения. Установите связь с отладчиком MICD2 командой *Debugger>Connect*. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (Operating System) и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger>Program*.

Активируйте выполнение программы в режиме реального времени. С помощью потенциометров изменяйте значения входных напряжений и фиксируйте результат преобразования на светодиодных индикаторах. Проконтролируйте правильность отображения информации. Результаты занесите в отчет.

Отключите отладчик системы MPLAB IDE командой *Debugger>Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от COM-порта компьютера.** Закройте проект командой *Project>Close*.

5.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма и исходный текст программы;
- результаты тестирования программы (временные диаграммы, значения временных интервалов);
- выводы.

5.4 Контрольные вопросы

5.4.1 Поясните по схеме алгоритма работу программы.

5.4.2 Как при автономной отладке можно проконтролировать корректность взаимодействия с модулем АЦП?

5.4.3 Как в среде симулятора можно имитировать выполнение аналогового преобразования?

6 Отладка программы вывода символьной информации

6.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки программы вывода символьной информации.

Объектом отладки в работе является программа, выполняющая запуск модуля АЦП, получение результатов преобразования, их обработку и вывод на 7-сегментные индикаторы в заданном формате. Для управления индикаторами в отладочном стенде применен специализированный контроллер MC14489BP. Эта микросхема обеспечивает вывод информации на 7- сегментные индикаторы в режиме динамического управления.

Информация от МК к специализированному контроллеру MC14489BP передается по последовательному синхронному интерфейсу, состоящему из трех линий:

- RE1 (DATA IN) – данные;
- RE0 (Clk) - синхроимпульсы;
- RB3 (-CS) – выборка кристалла.

Подпрограмма *Dysplay* обеспечивает формирование на линиях RE1, RE0, RB3 значений сигналов, необходимых для передачи информации. Поэтому объективный контроль работы этой подпрограммы должен выполняться путем проверки корректности формируемых последовательностей наборов этих сигналов. При выполнении автономной отладки для этой цели целесообразно использовать инструмент системы MPLAB IDE – логический анализатор. Логический анализатор использует информацию, формируемую в буфере трассы программ, и может ее интерпретировать в форме временных диаграмм сигналов на внешних выводах МК. Основное внимание в ниже следующем описании последовательности выполнения работы уделяется тестированию подпрограммы *Dysplay*. Контроль корректности других функций программы может быть выполнен при использовании приемов и инструментов, которые неоднократно применялись в лабораторных работах №1 – 5.

6.2 Описание последовательности выполнения

6.2.1 В папке PIC создайте свою рабочую папку. Скопируйте в нее подготовленный файл с исходным текстом программы *lr6.asm*. Создайте новый проект с помощью мастера *Project>Project Wizard*. Задайте имя проекта *lr6*, включите в проект исходный текст программы *lr6.asm*.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

6.2.2 Необходимо убедиться в правильности работы программы. Чтобы получить информацию об изменениях сигналов на выводах МК можно использовать логический анализатор отладчика системы MPLAB IDE. Запуск логиче-

ского анализатора выполняется по условию совпадения программного счетчика с заданным значением (адресом).

Тестирование подпрограммы Display:

- создайте окно наблюдения и включите в него ячейки Fig2, Fig1, Fig0;
- откройте окно логического анализатора командой *View>Logic Analyzer*, нажмите кнопку *Channels* и выберите интересующие сигналы на линиях RB3, RE0, RE1;
- установите переключатель *Trigger Position* на *Start*;
- установите точки останова на команде *call Display* и команде *return* подпрограммы *Dysplay*;
- запустите выполнение программы командой *Debugger>Run*;
- при первом останове программы, используя окно наблюдения, измените содержимое ячеек Fig2, Fig1, Fig0 на соответственно значения H'71', H'A5', H'5A';
- запустите программу;
- при втором останове программы скопируйте в буфер обмена временные диаграммы из окна логического анализатора по команде *Copy to Clipboard*;
- вставьте временные диаграммы из буфера обмена в отчет.

6.2.3 Подключите отладчик MICD2 к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику MICD2 источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика MICD2!

Активируйте отладчик MPLAB IDE для работы с MICD2 (команда *Debugger>Select Tool>MPLAB ICD2*). С помощью инструмента *MPLAB ICD2 Setup Wizard* настройте отладчик MPLAB IDE для выполнения комплексной отладки данного приложения. Установите связь с отладчиком MICD2 командой *Debugger>Connect*. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (Operating System) и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger>Program*.

Активируйте выполнение программы в режиме реального времени. С помощью потенциометра изменяйте значение входного напряжения и фиксируйте результат аналого-цифрового преобразования на 7-сегментных индикаторах. Проконтролируйте правильность отображения информации.

Отключите отладчик системы MPLAB IDE командой *Debugger>Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от СОМ-порта компьютера.** Закройте проект командой *Project>Close*.

6.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма программы;
- исходный текст программы;
- синтез управляющих слов;
- временные диаграммы сигналов;
- выводы.

6.4 Контрольные вопросы

6.4.1 Поясните по схеме алгоритма работу программы.

6.4.2 Каким образом выполняется вывод информации на 7-сегментные индикаторы отладочного стенда?

6.4.3 Как можно проконтролировать корректность работы подпрограммы вывода на 7-сегментные индикаторы?

6.4.4 Как выполняется настройка логического анализатора системы MPLAB IDE?

7 Отладка программы, использующей прерывания

7.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки в среде MPLAB IDE программ, использующих механизм прерываний. Объектом отладки в работе является программа, использующая механизм прерываний. Основные функции программы:

- отсчет времени и его отображение в секундах на 7-сегментных индикаторах;

- разнонаправленная модификация содержимого ячейки Cnt при нажатии с последующим отпусканием кнопок RB1 и RB2 и отображение ее содержимого на светоизлучающих диодах, подключенных к линиям порта PORTD.

Прерывания возникают при переполнении таймера TMR0, анализ состояния линий RB1 и RB2 выполняется путем программного опроса в главном цикле основной программы.

7.2 Описание последовательности выполнения

7.2.1 В своей рабочей папке создайте папку lr7. Скопируйте в нее подготовленный файл с исходным текстом программы lr7.asm. Создайте новый проект с помощью мастера *Project > Project Wizard*. Задайте имя проекта lr7, включите в проект исходный текст программы lr7.asm.

Откройте окно с исходным текстом программы. Выполните трансляцию программы (команда *Project>Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

7.2.2 Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки. Активируйте инструмент *MPLAB-SIM Simulator* командой *Debugger> Select Tool>MPLAB SIM*. Сформируйте удобную среду для отладки программы.

Контролировать текущие значения данных можно в окне наблюдения. Создайте смотровое окно командой *View> Watch*. Включите в смотровое окно основные ячейки:

- PORTB;
- CntT (счетчик переполнений таймера);
- Cnt (счетчик);
- Fig0 (разряды 1-0 7-сегментного индикатора);
- TMR0;
- INTCON;
- STATUS;
- W_temp;
- Status_temp.

Для автономной отладки программы необходимо иметь средства имитации нажатия кнопок RB1 и RB2. Для этих целей можно использовать асин-

хронные стимулы. Создайте асинхронные стимулы RB1 и RB2 командами меню *Debugger>Stimulus Controller> New Workbook*.

Контролировать сохранение точки возврата можно в окне стека. Откройте окно стека командой *View>Hardware Stack*. Для контроля формируемых интервалов времени откройте окно секундомера *Debugger>Stopwatch*.

7.2.3 Целесообразно сначала выполнить проверку работы основной программы, а затем проконтролировать возникновение и обработку прерываний.

Проверку работы основной программы удобно выполнять в режиме анимации (*Debugger>Animate*). Используя стимулы RB1 и RB2, проверьте корректность изменений содержимого ячейки Cnt.

Чтобы проконтролировать возникновение и обработку прерываний в режиме анимации, потребуется слишком много времени. Более рационально использовать режим прогона до точки останова. Установите точку останова на векторе прерывания. Запустите выполнение программы командой *Debugger>Run*. Убедитесь, что происходит передача управления на вектор прерывания. Зафиксируйте в отчете содержимое регистров TMR0, INTCON и стека. Выполните подпрограмму обработки прерывания в пошаговом режиме. Проконтролируйте, как происходит сохранение и восстановление контекста. По показаниям секундомера определите интервалы времени между последовательными прерываниями. Измените адрес точки останова так, чтобы можно было определить интервалы времени, через которые происходят изменения содержимого ячейки Fig0. Занесите результаты измерений в отчет.

7.2.4 Подключите отладчик MICD2 к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику MICD2 источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика MICD2!

Активируйте отладчик MPLAB IDE для работы с MICD2 (команда *Debugger>Select Tool>MPLAB ICD2*). С помощью инструмента *MPLAB ICD2 Setup Wizard* настройте отладчик MPLAB IDE для выполнения комплексной отладки данного приложения. Установите связь с отладчиком MICD2 командой *Debugger>Connect*. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (Operating System) и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger> Program*.

Активируйте выполнение программы в режиме реального времени. Нажатием кнопок RB1 и RB2 изменяйте состояния светодиодов. Убедитесь, что показания 7-сегментных индикаторов изменяются через одну секунду, независимо от состояния кнопок RB1 и RB2.

Отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от COM-порта компьютера.** Закройте проект командой *Project>Close*.

7.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- схему алгоритма и исходный текст программы;
- результаты тестирования программы;
- выводы.

7.4 Контрольные вопросы

7.4.1 Поясните по схеме алгоритма работу программы.

7.4.2 Когда целесообразно использовать механизм прерываний?

7.4.3 Как выполняется обработка запросов прерываний?

7.4.4 Как можно выполнять автономную отладку программ, использующих механизм прерываний?

8 Отладка программной реализации микропрограммного автомата

8.1 Описание работы с кратким изложением теории

Цель работы - получить практические навыки выполнения автономной и комплексной отладки в среде MPLAB IDE программы, реализующей функцию микропрограммного управляющего автомата.

Объектом отладки в работе является программа, которая:

- опрашивает входные линии RE2 и RA5 МК, на которые поступает набор входных сигналов $X = \langle x_2, x_1 \rangle$;
- формирует на линиях 7-4 PORTD текущие значения набора выходных сигналов $Y = \langle y_4, y_3, y_2, y_1 \rangle$.

Смена внутренних состояний автомата происходит при прерывании, инициатором которого является положительный фронт внешнего сигнала Clk на линии INT/PB0.

Перед выполнением тестирования программы, необходимо составить его план. Для автомата Мура, алгоритм которого задан ГСА и графом состояний (рисунок 8.1), предусмотрено выполнение переходов:

- $s_0 - s_1$; $s_0 - s_2$;
- $s_1 - s_3$; $s_1 - s_5$;
- $s_2 - s_3$; $s_2 - s_5$;
- $s_3 - s_4$;
- $s_4 - s_0$;
- $s_5 - s_3$; $s_5 - s_5$.

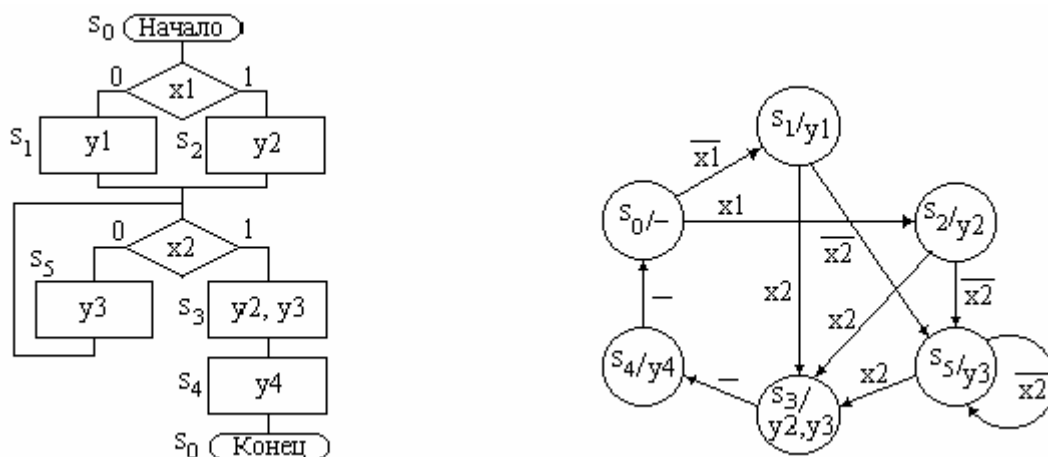


Рисунок 8.1

Необходимо сформировать такую цепочку из этих переходов, чтобы в ней были все переходы, а их общее число было минимально, например:

$s_0 - s_1 - s_5 - s_5 - s_3 - s_4 - s_0 - s_2 - s_3 - s_4 - s_0 - s_2 - s_5$.

Для данной последовательности сформируем тест, задавая последовательность соответствующих значений входных сигналов x_1 , x_2 , необходимых для выполнения каждого перехода:

x_1 0 – 0 – 0 – 0 – 0 – 1 – 1 – 1 – 1 – 1 – 0
 x_2 0 – 0 – 0 – 1 – 1 – 1 – 1 – 1 – 1 – 1 – 0 – 0

Каждый набор входных сигналов должен сопровождаться синхроимпульсом Clk.

8.2 Описание последовательности выполнения

8.2.1 В своей рабочей папке создайте папку lr8. Скопируйте в нее подготовленный файл с исходным текстом программы lr8.asm. Создайте новый проект с помощью мастера *Project > Project Wizard*. Задайте имя проекта lr8, включите в проект исходный текст программы lr8.asm.

Откройте окно с исходным текстом программы. Автономную отладку программы целесообразно выполнять с отключенной функцией подавления дребезга контактов. Выполните трансляцию программы (команда *Project > Make*). Если программа содержала синтаксические и другие формальные ошибки, исправьте их и повторите трансляцию.

8.2.2 Проверку работы программы, поиск и исправление возможных ошибок выполняют в ходе отладки. Активируйте инструмент *MPLAB-SIM Simulator* командой *Debugger > Select Tool > MPLAB SIM*. Сформируйте удобную среду для отладки программы. Для отладки программы необходимо иметь средства имитации источников входных сигналов $X = \langle x_1, x_2 \rangle$ и сигнала Clk. Для этих целей можно использовать стимулы. Также необходимы инструменты для отслеживания выполнения программы.

Автономную отладку данной программы можно выполнить в режиме анимации или режиме автоматического прогона программы. В первом случае потребуются асинхронные стимулы для сигналов на линиях RE2, RA5 и RB0. Контролировать текущие значения данных можно в окне наблюдения, в котором должны отражаться ячейки: PORTA, PORTB, PORTD, PORTE, InWord, OutWord. Методика отладки программ в этом режиме подробно рассмотрена на лабораторном занятии №4.

Проще и быстрее автономную отладку данной программы выполнить в режиме автоматического прогона при фиксации основных ячеек в буфере трассы и отображении входных и выходных сигналов управляющего автомата в окне логического анализатора. В данном случае потребуется использовать синхронные стимулы.

Создайте стимулы RE2, RA5 и RB0 командами меню *Debugger > Stimulus Controller > New Workbook*. В окне *Stimulus* выберите закладку *Clock Stimulus*, на которой задайте параметры стимула тактового сигнала Clk с периодом повто-

рения 100 машинных циклов на линии RB0. Соответствующие значения параметров приведены в таблице 8.1.

Таблица 8.1

Label	Pin	Initial	Low Cyc	High Cyc	Begin	End
Clk	RB0	Low	50	50	At Start	Never

Для рассматриваемого примера, при условии следования входных наборов с интервалом 100 циклов, соответствующие изменения сигнала x_1 на линии RA5 должны происходить на отметках времени $6 \cdot 100 = 600$ циклов и $(6+5) \cdot 100 = 1100$ циклов. Изменения сигнала x_2 на линии RE2 должны происходить на отметках времени $3 \cdot 100 = 300$ циклов и $(3+7) \cdot 100 = 1000$ циклов.

На закладке *Pin/Register Actions* нажмите на заголовок столбца таблицы *Click here to Add Signals*. В открывшемся окне последовательно выберите нужные сигнальные линии и добавьте их в список *Secected Signals* нажатием кнопки *Add*. Подтвердите выбор нажатием кнопки *OK*. В ячейках сформированной таблицы задайте параметры стимулов сигналов x_1 , x_2 , действующих на линиях RE2 и RA5. Для рассматриваемого примера образец таблицы приведен на рисунке 8.2. Нажатием кнопки *Apply* подтвердите подключение созданных стимулов к программно-логической модели. Сохраните стимулы в файле (кнопка *Save*).

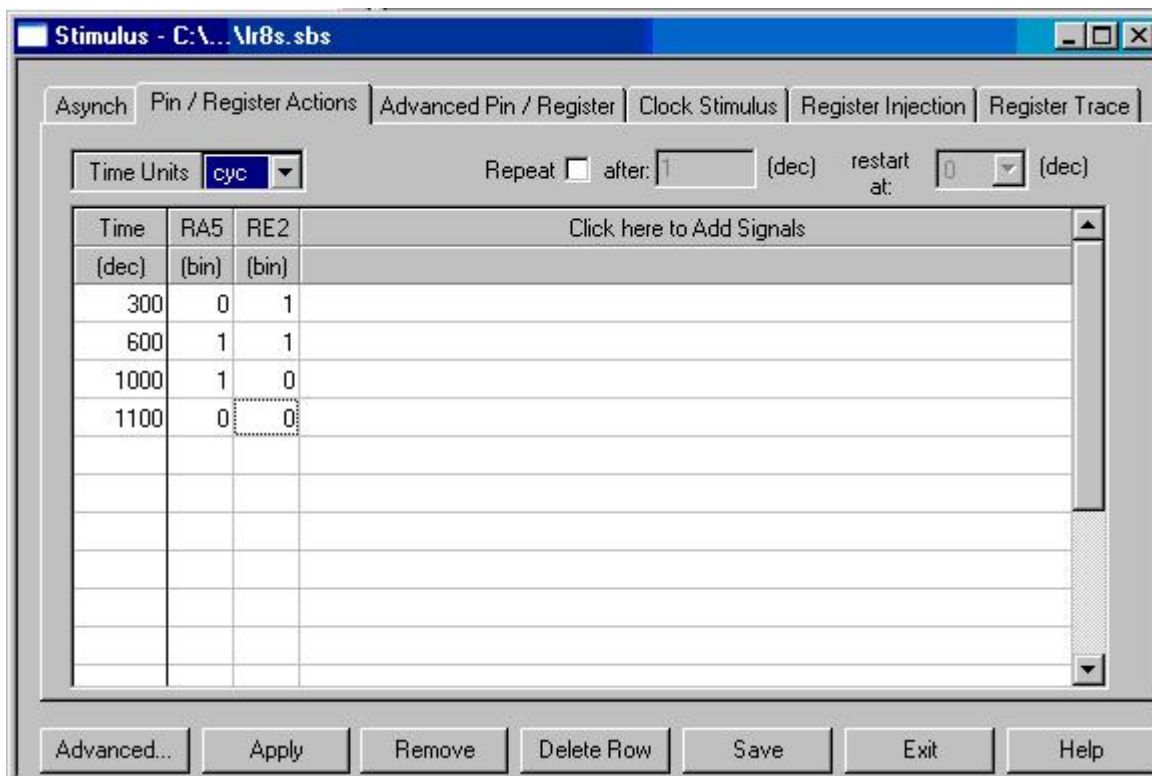


Рисунок 8.2

8.2.3 Выполните тестирование программы, используя подготовленные стимулы. В соответствии с планом тестирования необходимо выполнять программу на протяжении следования 12 наборов входных сигналов. Это значит, что на протяжении интервала выполнения программы должно произойти 12 прерываний программы по фронту сигнала Clk. Прекратить выполнение программы можно на 13-м по счету прерывании. Наиболее просто это обеспечить, используя сложные условия останова отладчика системы MPLAB IDE. Выполните команду *Debugger>Complex Breakpoints*. В открывшемся окне *Simulator Complex Breakpoints* нажмите кнопку *Add Breakpoint*. В окне *Set Breakpoint* на закладке *Program Memory* необходимо в окне *Address* задать метку точки останова – вход в подпрограмму обработки прерывания. В окне *Pass Count* выберите условие *Event must occur Count times*, в окне *Count* задайте число событий (прерываний) до останова программы.

Откройте окно логического анализатора командой *View>Logic Analyzer*, нажмите кнопку *Channels* и выберите сигналы на линиях RE2, RA5, RB0, RD4, RD5, RD6, RD7. Установить переключатель *Trigger Position* на *Start*.

Запустите выполнение программы в автоматическом режиме. После останова программы, используя окно логического анализатора, исследуйте временные диаграммы зарегистрированных сигналов.

Убедитесь, что:

- происходит смена значений выходных сигналов на линиях RD4, RD5, RD6, RD7 после формирования положительного фронта тактового сигнала автомата на линии RB0;

- смена формируемых выходных наборов соответствует заданному алгоритму функционирования автомата.

Зарегистрированные временные диаграммы для рассматриваемого примера приведены на рисунке 8.3.

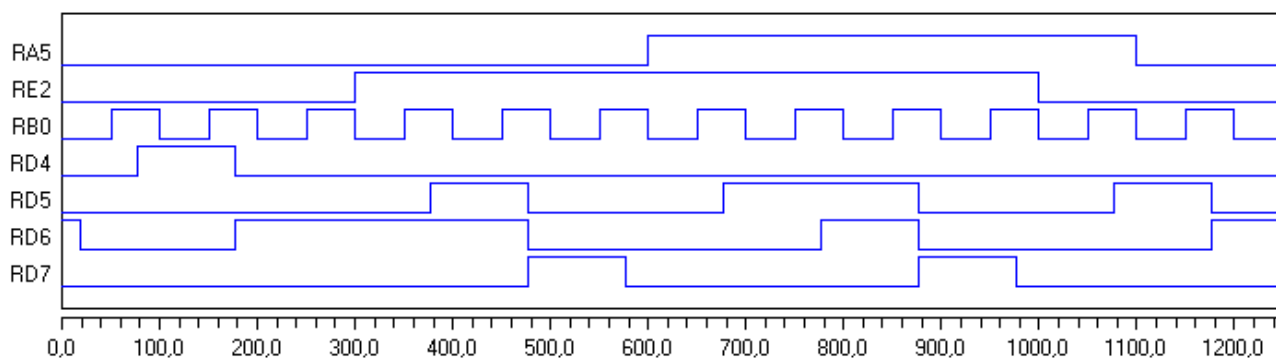


Рисунок 8.3

При успешном прохождении теста скопируйте временные диаграммы из окна логического анализатора в буфер обмена по команде *Copy to Clipboard*. Вставьте временные диаграммы из буфера обмена в отчет.

В случае, если значения выходных сигналов не изменяются, необходимо проверить настройку системы прерываний МК и корректность работы подпрограммы обработки прерываний по рекомендациям, рассмотренным на лабораторном занятии №7. Если характер изменений значений выходных сигналов отличается от правильных, то причина этого, скорее всего, в ошибках, допущенных при формировании табличного представления функций переходов выходов автомата. После локализации и исправления ошибок, необходимо повторить трансляцию и тестирование исправленного варианта программы.

8.2.4 Перед выполнением комплексной отладки программы необходимо подключить функцию подавления дребезга контактов кнопки и повторно выполнить трансляцию программы.

Подключите отладчик MICD2 к стенду и персональному компьютеру. **Только после этого можно подключить к отладчику MICD2 источник питания.** Нарушение этого требования может привести к выходу из строя портов компьютера и отладчика MICD2!

Активируйте отладчик MPLAB IDE для работы с MICD2 (команда *Debugger>Select Tool>MPLAB ICD2*). С помощью инструмента *MPLAB ICD2 Setup Wizard* настройте отладчик MPLAB IDE для выполнения комплексной отладки данного приложения. Установите связь с отладчиком MICD2 командой *Debugger>Connect*. Если в окне *Output* на закладке *MPLAB ICD2* появится сообщение об ошибке, проверьте подключение источника питания, разъемы и соединительные кабели.

После установления связи, загрузки программного обеспечения отладчика (Operating System) и самотестирования выполните загрузку целевого приложения в микроконтроллер отладочного стенда по команде *Debugger> Program*.

Активируйте выполнение программы в режиме реального времени. Устанавливая DIP-переключателями необходимые комбинации входных сигналов $X = \langle x_1, x_2 \rangle$ и нажатием кнопки RB0 имитируя формирование фронта сигнала Clk, проконтролируйте по состоянию светодиодов значения выходных сигналов и кода состояния. Убедитесь, что работа программной реализации управляющего микропрограммного автомата соответствует заданному алгоритму.

Отключите отладчик системы MPLAB IDE командой *Debugger> Select Tool>None*. **Отключите питание MICD2. После этого можно отключить кабель от COM-порта компьютера.** Закройте проект командой *Project>Close*.

8.3 Оформление работы

Отчет по лабораторной работе должен содержать:

- граф состояний автомата;
- схему алгоритма и исходный текст программы;
- результаты тестирования программы (временные диаграммы);
- выводы.

8.4 Контрольные вопросы

8.4.1 Поясните по схеме алгоритма работу программы.

8.4.2 Как можно сформировать входной тест для проверки микропрограммного автомата?

8.4.3 Как можно выполнять автономную отладку программной реализации управляющего микропрограммного автомата?

8.4.4 Как можно выполнять комплексную отладку программной реализации управляющего микропрограммного автомата?

Список использованных источников

1 MPLAB IDE [Электронный ресурс]: User's Guide DS51519B.– Microchip, 2006. – 277 р. – Режим доступа: WWW.URL: [http: // www.microchip.com/](http://www.microchip.com/). – 27.06.2008.

2 MPASM [Электронный ресурс]: Руководство пользователя. – М.: ООО «Микро-Чип», 2001. – 183 с. – Режим доступа: WWW.URL: [http: // www.microchip.ru/](http://www.microchip.ru/). – 27.06.2008.

3 PIC16F87X [Электронный ресурс]: Техническое описание. – М.: ООО «Микро-Чип», 2001. – 61 с. – Режим доступа : WWW.URL : [http: // www.microchip.ru/](http://www.microchip.ru/). – 27.06.2008.

4 MPLAB ICD2 [Электронный ресурс]: User's Guide DS51331C.– Microchip, 2007. – 138 р. – Режим доступа: WWW.URL: [http: // www.microchip.com/](http://www.microchip.com/). – 27.06.2008.