

1. ПРАКТИЧЕСКАЯ РАБОТА № 1 «ОСНОВЫ РАБОТЫ С ПРОГРАММНЫМ ОБЕСПЕЧЕНИЕМ MULTISIM»

Цель работы: знакомство с интерфейсом программы Multisim; изучение создания проекта и программного файла; изучение системы команд МК 8051.

1.1. Создание проекта в программе Multisim

Для запуска программы выберите в разделе *Все программы National Instruments – Circuit Design Suite 12.0 – Multisim 12.0*. После запуска программы откроется главное окно, основные элементы которого приведены на рис. 1.1. При открытии программы Multisim автоматически создается проект схемы под названием Design1.

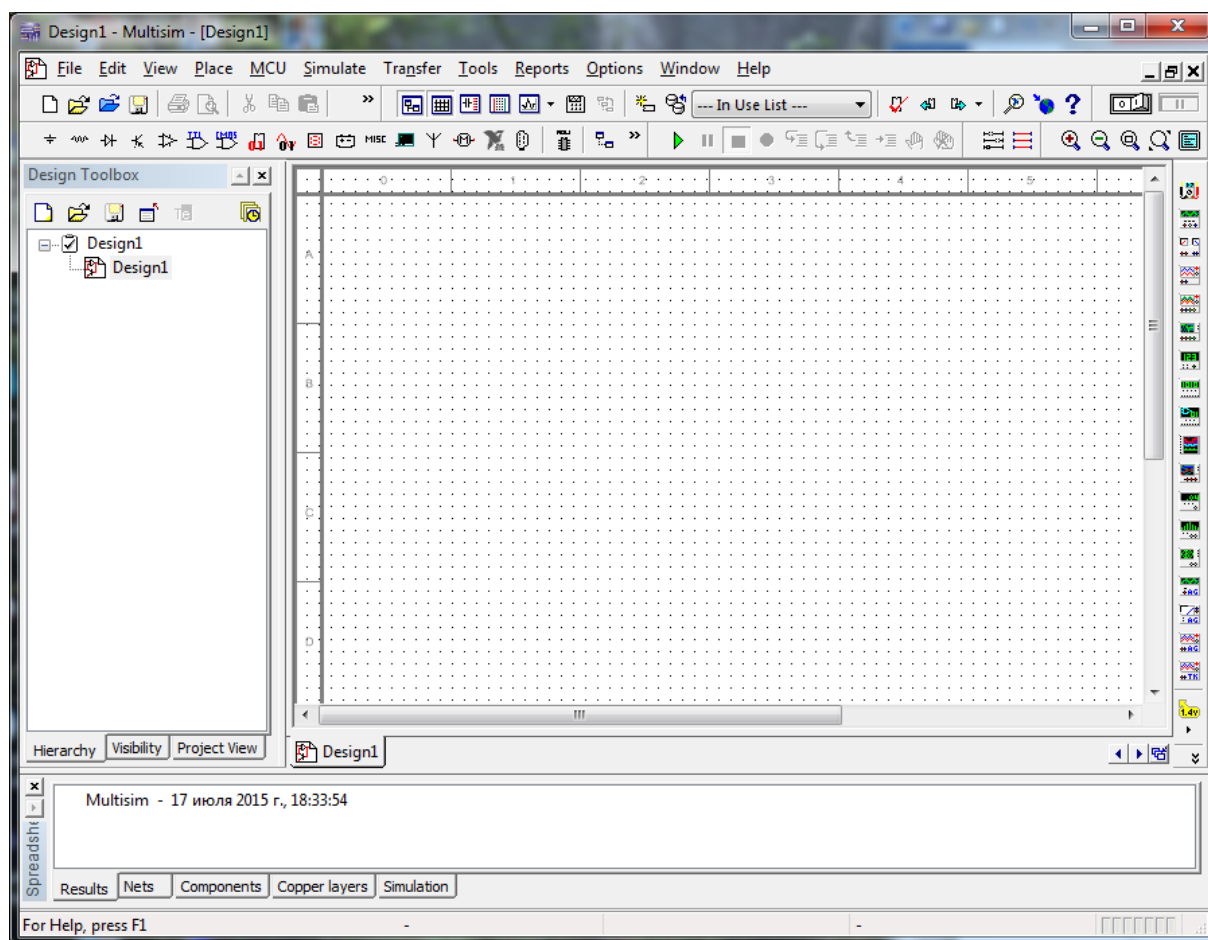


Рис. 1.1. Основные элементы пользовательского интерфейса

Основными элементами пользовательского интерфейса являются: главное меню, панель инструментов, строка состояния, полосы прокрутки и другие стандартные элементы окна программы Windows.

Главное меню программы Multisim обладает большим набором инструментов для подготовки схемы и проведения анализа. Пункты главного меню *File*, *Edit*, *View* являются обычными меню с набором команд для работы с файлами и проектами, печати (*File*), редактирования и изменения свойств чертежа, ориентирования, удаления, выделения, перемещения элементов схем (*Edit*), настройки пользовательского интерфейса с возможностью изменения набора инструментальных панелей, масштабирования рабочей области (*View*).

Пункт меню *Place*:

Component... – предназначен для выбора и размещения компонентов схемы;

Wire – нанесение соединительных проводов;

Bus – построение информационных шин;

Junction – определение электрических узлов-соединителей;

Comment – определение комментария в схеме;

Connectors – введение соединителей;

Text – введение пояснительного текста;

Graphics – вставка элементарных графических элементов и графических изображений из внешнего файла;

Title Block – задание параметров штампа для схемы.

Пункт меню *MCU* позволяет записать программный код для работы выбранного микропроцессора и отладить программу.

Пункт меню *Simulate* задает типы проводимого анализа, позволяет выбрать встроенные приборы, сохранить результаты и провести их обработку. В этом меню можно изменить временной шаг моделирования при помощи пункта *Interactive Simulation Settings* (рис. 1.2).

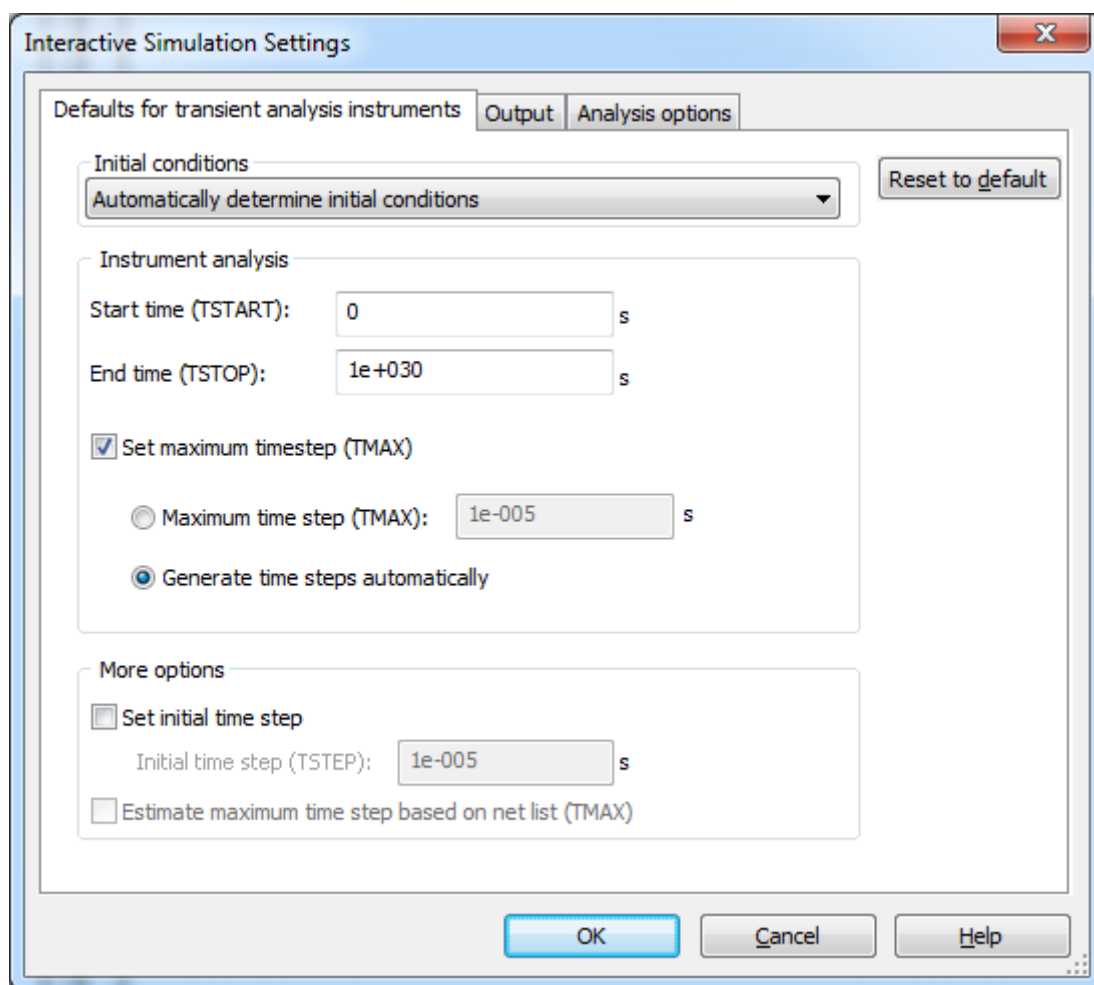


Рис. 1.2. Окно настройки временного шага моделирования

В программе по умолчанию выбрана автоматическая генерация шага моделирования (10^{-5} с). Для изменения временного шага моделирования нужно активировать строчку Maximum time step (TMAX) и указать необходимый шаг.

Пункт меню *Transfer* – здесь можно передать исходные данные для трассировки печатных плат в программу Ultiboard.

Меню *Tools* позволяет работать с базой данных компонентов программы, использовать возможности автоматизированного проектирования, мастера проектирования имеющихся типовых устройств на основе таймера 555 серии, полосовых фильтров, операционных усилителей и каскадов усиления на биполярных транзисторах, для которых можно выбрать требуемые выходные параметры.

В этом меню также можно подобрать исходные данные для многовариантного анализа исследуемой схемы, проверить схему на ошибки, редактировать имена электронных компонентов и даже реализовать дальнейшее сохранение полученной картинки в виде графического файла.

Пункт меню *Reports* предлагает детальный отчет о схеме: числе и типе компонентов, их параметрах, сведения об узлах схемы и многое другое. Кроме этого полученную информацию можно передать в офисные программы для дальнейшего использования.

Пункт меню *Option* задает условия работы по подготовке и сохранению схемы, определения внешнего вида схемы и условий ввода и размещения элементов при рисовании. Он содержит следующие подменю: *Global Preferences*, *Sheet Properties* и *Customize Interface*.

Подменю *Global Preferences* определяет режимы и условия работы программы, как в процессе ввода схемы, так и при сохранении введенной схемы в виде файла. Так, на закладке *Paths* дается путь до папок хранения файлов схем, файлов конфигурации и баз данных.

На закладке *Save* определяются режимы сохранения и размеры файла данных.

На закладке *Components* определяются действия программы при выборе и установке компонента в рабочую область схемы, выбор стандарта вида компонента, параметры автоматизации измерения результатов анализа, параметры моделирования цифровых устройств.

На закладке *General* определяются действия при движении колесика мыши, задаются возможности автоматизации соединения проводников и возможности выбора выделяемых движением курсора областей.

Подменю *Preview* определяет внешний вид подготовленной, нарисованной схемы.

В подменю *Message Prompts* можно выбрать, в каких ситуациях будут выводиться сообщения.

В разделе *Simulation* задаются параметры режима моделирования. Внешний вид окон моделирования, сообщения об ошибках.

Пункт меню *Window* – стандартный, осуществляет навигацию между внутренними окнами программы и регулирует расположение этих окон. Аналогичное можно сказать и о пункте меню *Help*.

Сохраните проект под другим именем. Для этого выберите пункт меню *File -> Save As*, в открывшемся диалоговом окне укажите новое название схемы и место расположения файла. Обратите внимание, чтобы в директории, где будет располагаться файл и в названии файла использовались только английские буквы. Сохранять схемы в папки с русскими названиями нельзя.

1.2. Размещение элементов на схеме и создание программного файла

Элементы схемы выбираются из базы данных и размещаются на рабочем поле тремя способами:

- 1) Через Главное меню (*Place – Component...*);
- 2) Через контекстное меню рабочей области (*Place Component...*);
- 3) Через панель компонентов (рис. 1.3).

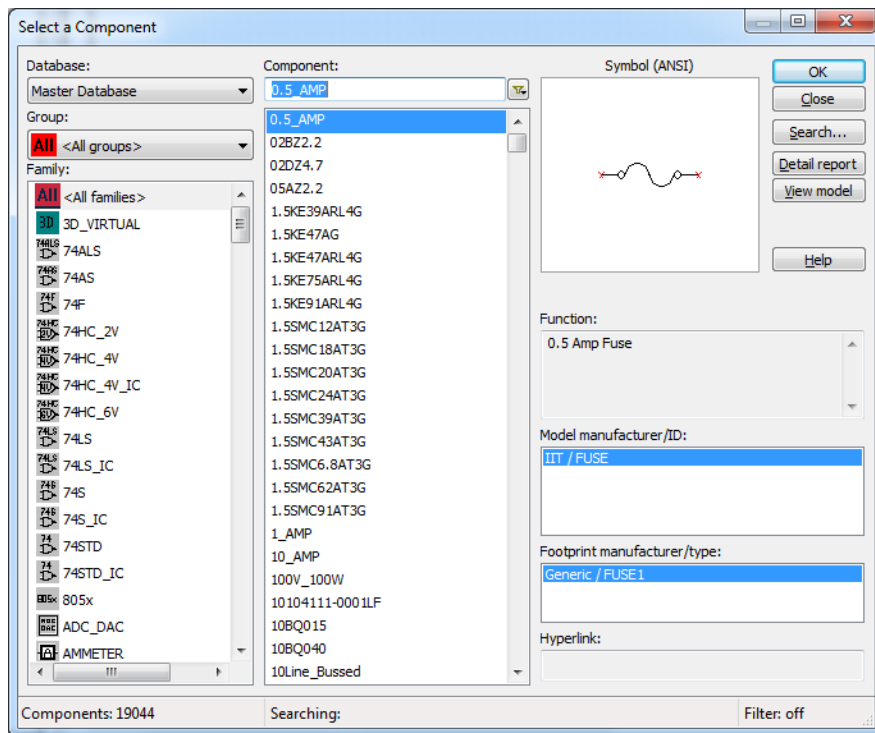


Рис. 1.3. Панель компонентов

Имеются следующие группы элементов:

Группа Sources – источники энергии и сигналов

POWER_SOURCES – источники питания (AC – источник питания переменного тока, DC – источник питания постоянного тока, DGND – цифровая земля, GROUND – аналоговая земля, VCC, VDD – положительное напряжение питания, VEE, VSS – отрицательное напряжение, земля).

Аналоговое заземление используется во всех процессах моделирования за исключением моделирования цифровых устройств в реальном времени (в этом режиме, кроме задержки в логическом элементе, моделируется время фронта и время спада, выходное напряжение имеет сглаженные фронты, а сам процесс моделирования требует большего времени).

SIGNAL_VOLTAGE_SOURCES – сигнальные источники напряжения.

SIGNAL_CURRENT_SOURCES – сигнальные источники тока.

CONTROLLED_VOLTAGE_SOURCES – регулируемые источники напряжения.

CONTROLLED_CURRENT_SOURCES – регулируемые источники тока.

CONTROL_FUNCTION_BLOCKS – функциональные блоки управления.

DIGITAL_SOURCES – цифровые источники.

Группа Basic – группа с базовыми элементами

RPACK – резистивная сборка.

SWITCH – переключатели (ключи, PB_DPST – кнопка, блоки с 2–10 переключателями).

TRANSFORMER – трансформатор.

Z_LOAD – переменная нагрузка.

RELAY – реле.

SOCKETS – сокет.

RESISTOR – резистор.

CAPACITOR – конденсатор.

INDUCTOR – катушки индуктивности.

CAP_ELECTROLIT – электролитические конденсаторы.

VARIABLE_CAPACITOR – переменные конденсаторы.

VARIABLE_INDUCTOR – переменные катушки индуктивности.

POTENTIOMETER – потенциометры.

Группа Diodes – диоды

DIODE – диод.

ZENER – стабилитроны (диоды Зенера).

LED – светодиоды.

FWB – диодные мосты.

SCHOTTKY_DIODE – диоды Шоттки.

SCR – тиристоры триодные, запираемые в обратном направлении с управлением по катоду.

DIAC – диоды двунаправленные.

TRIAC – тиристоры триодные симметричные (двунаправленные).

VARACTOR – варикапы (емкостные диоды).

PIN_DIODE – pin диоды (содержат область собственной проводимости между сильнолегированными областями).

Группа Transistors – транзисторы

BJT NPN – биполярные транзисторы типа NPN.

BJT PNP – биполярные транзисторы типа PNP.

MOS_ENH_N – транзистор канальный с встроенным каналом N-типа.

MOS_ENH_P – транзистор канальный с индуцированным каналом P-типа.

JFET N – транзистор полевой с проводимостью типа N.

JFET P – транзистор полевой с проводимостью типа P.

POWER MOS N – мощный канальный транзистор с каналом N-типа.

POWER MOS P – мощный канальный транзистор с каналом P-типа.

POWER MOS COM – мощный канальный транзистор (комплементарная технология).

UJT – тиристор триодный, запираемый в обратном направлении с управлением по аноду.

THERMAL MODELS – температурные модели.

Группа Analog – аналоговые компоненты

OPAMP – операционные усилители.

COMPARATOR – компараторы.

WIDEBAND AMPS – широкополосные усилители.

SPECIAL FUNCTION – компоненты, реализующие специальные функции.

Группа TTL – элементы транзисторно-транзисторной логики 74-серии

Группа CMOS – комплементарная МОП-структура (комплементарные транзисторы)

Группа MCU

Микроконтроллеры – МК (805х, PIC).

Микросхемы памяти – RAM, ROM (НМ-65642-883 (8К x 8), НМ61116А120(2К x 8)), ПЗУ (27С128-12L(16К x 8), 27С256-15L (32К x 8)), ППЗУ (27С64Q350-883)).

Группа Advanced peripherals

Усовершенствованные периферийные устройства, такие как виртуальная цифровая клавиатура (4x4, 4x5), LCD-дисплей, светофор.

Группа Misc digital – различные цифровые микросхемы

MEMORY – микросхемы памяти.

LINE TRANSCEIVER – линейные приемопередатчики.

Группа Mixed – устройства смешанного сигнала

ANALOG SWITCH – аналоговые переключатели.

TIMER – таймер.

ADC_DAC – АЦП (ADS8364Y, AD16), ЦАП (DAC7643_FP32).

MULTIVIBRATORS – мультивибраторы.

Группа Indicators – индикаторы

VOLTMETER – вольтметры.

AMMETER – амперметры.

PROBE – пробники.

BUZZER – автоматические прерыватели.

LAMP – лампы.

HEX DISPLAY – дисплеи (светоиндикаторы – 15-сегментные, семисегментные, с общим катодом, с общим анодом, с десятичной точкой, без точки, дисплеи с «+» или «-», 7 сегментные дисплеи с двумя цифрами (с десятичной точкой, с общим анодом или с общим катодом).

BARGRAPH – столбцовая диаграмма.

Группа Power – компоненты, относящиеся к источникам питания и связанные с ними:

FUSE – плавкие предохранители.

VOLTAGE REFERENCE – источники опорного напряжения.

VOLTAGE REGULATOR – потенциометры.

POWER_CONTROLLERS – контроллеры источников питания

Группа Misc – прочее

OPTOCOUPLER – оптроны.

CRYSTAL – кварцевые резонаторы.

VACUUM TUBE – электронные лампы.

BOOST CONVERTER – усилители-преобразователи.

LOSSY TRANSMISSION LINE – линия передачи с потерями.

LOSSLESS LINE TYPE1 – линия без потерь, тип 1.

LOSSLESS LINE TYPE1 – линия без потерь, тип 2.

FILTERS – фильтры.

NET –сеть связи.

Группа RF – радиочастотные устройства

RF CAPACITOR – радиочастотные конденсаторы.

RF INDUCTOR – радиочастотные катушки индуктивности.

RF BJT NPN – радиочастотные биполярные транзисторы типа NPN.

RF BJT PNP – радиочастотные биполярные транзисторы типа PNP.

RF MOS 3TDN – радиочастотные полевые транзисторы с встроенным каналом N-типа.

TUNNEL DIODE – радиочастотные туннельные диоды.

STRIP LINE – полосковые линии.

Группа Electro Mechanical – электромеханические устройства

SENSORS – сенсорные переключатели.

MOMENTARY SWITCHES – мгновенные переключатели.

SUPPLEMENTARY SWITCHES – дополнительные переключатели.

TIMED CONTACTS – синхронизированные контакты.

COILS RELAYS – реле.

MECHANICAL_LOADS – механические нагрузки.

PROTECTION DEVICES – элементы защиты (предохранители).

Все рассмотренные выше компоненты являются реальными (промышленными) и имеют определенные, неизменяемые значения параметров. В схеме по умолчанию они обозначаются синим цветом. В базе данных есть также и виртуальные компоненты, имеющие в своем названии приставку VIRTUAL (неявно отображаются на схеме черным цветом). Виртуальные компоненты необходимы для исследований, так как пользователь может назначить им произвольные значения параметров.

Основным элементом любой микропроцессорной системы является микропроцессор. В данной работе используется микроконтроллер i8051 семейства MCS-51.

Семейство MCS-51 включает более 200 модификаций 8-разрядных микроконтроллеров с широким спектром периферии, которые имеют

общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными характеристиками микроконтроллеров i8051 являются:

- внутренне ОЗУ объемом 128 байт;
- четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;
- два 16-разрядных таймера-счетчика;
- встроенный тактовый генератор;
- адресация 64 Кбайт памяти программ и 64 Кбайт памяти данных;
- две линии запросов на прерывание от внешних устройств;
- интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

На рис. 1.4 приведено назначение выводов МК 8051.

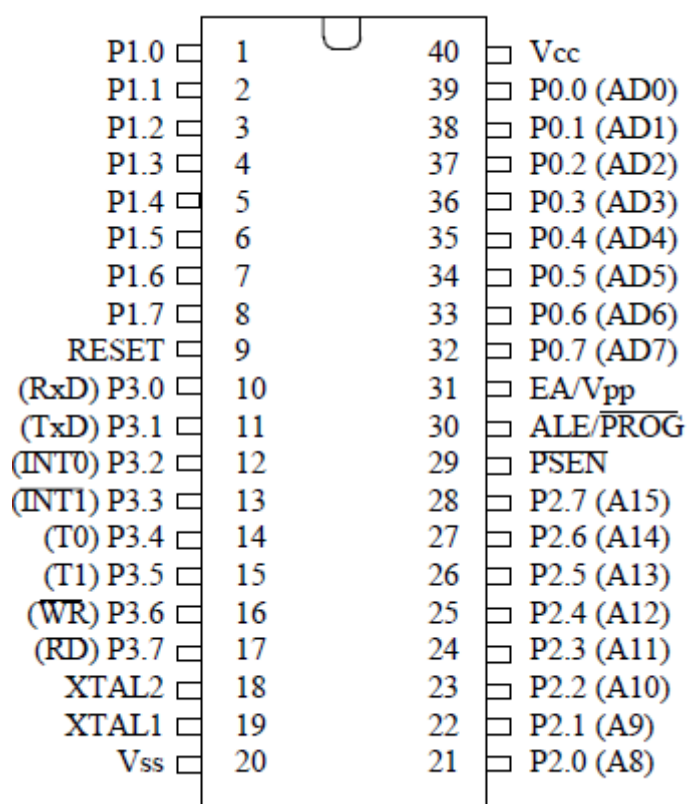


Рис. 1.4. Назначение выводов 8051

Vss — потенциал общего провода ("земли");

U_{cc} — основное напряжение питания +5 В;

X1,X2 — выводы для подключения кварцевого резонатора;

RST — вход общего сброса микроконтроллера;

PSEN — разрешение внешней памяти программ; выдается только при обращении к внешнему ПЗУ;

ALE — строб адреса внешней памяти;

EA — отключение внутренней программной памяти; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнее ПЗУ; игнорируя внутреннее (если последнее имеется);

P1 — восьмибитный квазидвунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;

P2 — восьми битный квазидвунаправленный порт, аналогичный P1; кроме того, выводы этого портаиспользуются для выдачи адресной информации при обращении к внешней памяти программ илиданных (если используется 16-битовая адресация);

P3 — восьми битный квазидвунаправленный порт, аналогичный P1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных;

P0 — восьми битный двунаправленный порт ввода-вывода информации: при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных.

Все порты МК имеют возможность побитовой адресации разрядов и являются двунаправленными, причем имеется возможность в каждом порту часть разрядов использовать для ввода данных, а часть для вывода.

Каждый из портов содержит регистр-защелку (SFR P0 — SFR P3), выходную цепь и входной буфер. Каждый из разрядов регистра-защелки

SFR является D-триггером, информация в который заносится с внутренней шины данных микроконтроллера по сигналу «Запись в SFR P_x» (x= 0, 1, 2, 3) от центрального процессора (ЦП). С прямого выхода D-триггера информация может быть выведена на внутреннюю шину по сигналу «Чтение SFR P_x» от ЦП, а с вывода микросхемы по сигналу «Чтение выводов P_x». Одни команды активизируют сигнал «Чтение SFR P_i», другие - «Чтение выводов P_i».

Для перевода любой линии портов P1 — P3 в режим ввода информации необходимо в соответствующий разряд SFR занести 1.

Корпус МК имеет небольшое количество выводов, поэтому помимо основных функций они выполняют еще ряд альтернативных функций. Порты P0 и P2 используются при обращении к внешней памяти. При этом на выходах P0 младший байт адреса внешней памяти мультиплексируется с вводимым/выводимым байтом. Выходы P2 содержат старший байт адреса внешней памяти, если адрес 16-разрядный. При использовании восьмиразрядного адреса портом P2 можно пользоваться для ввода-вывода информации обычным образом. При обращении к внешней памяти в P0 автоматически заносятся 1 во все биты. Информация в P2 при этом остается неизменной. Порт P3 помимо обычного ввода и вывода информации используется для формирования и приема специальных управляющих и информационных сигналов. Разряды порта (все или частично) при этом могут выполнять альтернативные функции приведенные в табл. 1.1.

Таблица 1.1

Функции выводов порта P3

Вывод порта	Альтернативная функция
P3.0	RXD - вход последовательного порта
P3.1	TXD - выход последовательного порта
P3.2	INT0 - внешнее прерывание 0

P3.3	INT1 - внешнее прерывание 1
P3.4	T0 - вход таймера-счетчика 0
P3.5	T1 - вход таймера-счетчика 1
P3.6	WR - строб записи во внешнюю память данных
P3.7	RD - строб чтения из внешней памяти данных

У МК 8051 память программ и память данных являются самостоятельными и независимыми друг от друга устройствами, адресуемыми различными командами и управляющими сигналами. Объем встроенной памяти программ – 4 Кбайт, расположенной на кристалле памяти данных – 128 байт. Помимо встроенной памяти имеется возможность подключить внешнюю память программ и данных объемом 64 Кбайт.

Первые 32 байта адресного пространства данных организованы в виде четырех банков регистров общего назначения, каждый из которых состоит из восьми регистров R0-R7. В любой момент программе доступен только один банк регистров. Выбор банка регистров осуществляется установкой соответствующих бит слова состояния программы PSW.

К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register). Доступные регистры SFR приведены в табл. 1.2.

Таблица 1.2

Регистры SFR

Символ	Наименование
*ACC	Аккумулятор (Accumulator)
*B	Регистр расширитель аккумулятора (Multiplication Register)
*PSW	Слово состояния программы (Program Status Word)

*P0	Порт 0 (SFR P0)
*P1	Порт 1 (SFR P1)
*P2	Порт 2 (SFR P2)
*P3	Порт 3 (SFR P3)
SP	Регистр указатель стека (Stack Pointer)
DPH	Старший байт регистра указателя данных DPTR (Data Pointer High)
DPL	Младший байт регистра указателя данных DPTR (Data Pointer Low)
TH0	Старший байт таймера 0 ()
TL0	Младший байт таймера 0 ()
TH1	Старший байт таймера 1 ()
TL1	Младший байт таймера 1 ()
TMOD	Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register)
*TCON	Регистр управления статуса таймеров (Timer/Counter Control Register)
*IP	Регистр приоритетов (Interrupt Priority Control Register)
*IE	Регистр маски прерывания (Interrupt Enable Register)
PCON	Регистр управления мощностью (Power Control Register)
*SCON	Регистр управления приемопередатчиком (Serial Port Control Register)
SBUF	Буфер приемопередатчика (Serial Data Buffer)

Регистры специальных функций управляют работой блоков, входящих в микроконтроллер.

Регистры-защелки SFR параллельных портов P0...P3 - служат для ввода-вывода информации.

Две регистровые пары с именами TH0, TL0 и TH1, TL1 представляют собой регистры, двух программно-управляемых 16-битных таймеров-счетчиков.

Режимы таймеров-счетчиков задаются с использованием регистра TMOD, а управление ими осуществляется с помощью регистра TCON.

Для управления режимами энергопотребления микро-ЭВМ используется регистр PCON.

Регистры IP и IE управляют работой системы прерываний микро-ЭВМ, регистры SBUF и SCON — работой приемопередатчика последовательного порта.

Регистр-указатель стека SP в микро-ЭВМ рассматриваемого семейства — восьмибитный. Он может адресовать любую область внутренней памяти данных. В отличие от микропроцессора KP580BM80, МК семейства 8051 стек «растет вверх», т.е. перед выполнением команды PUSH или CALL содержимое SP инкрементируется, после чего производится запись информации в стек. Соответственно при извлечении информации из стека регистр SP декрементируется после извлечения информации. В процессе инициализации МК после сигнала сброса или при включении питающего напряжения в SP заносится код 07H. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08H.

Регистр-указатель данных DPTR чаще всего используют для фиксации 16-битного адреса в операциях обращения к внешней памяти программ и данных. С точки зрения программиста он может выступать как в виде одного 16-битного регистра, так и в виде двух независимых регистров DPL и DPH

Аккумулятор (ACC) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с

использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п.

При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW.

Регистр В используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру SFR, производится аналогично аккумулятору.

1.3. Создание проекта и программного файла

Добавьте на рабочую область микроконтроллер i8051, который расположен в БД Multisim в группе MCU. При этом микроконтроллер (МК) можно выбрать двумя способами:

- через пункт меню *Place – Component – MCU – 805x – 8051*;
- через панель компонентов: – *805x – 8051*.

После установки микроконтроллера на рабочей области, появляется всплывающее окно «Мастер по созданию программного файла», которое предлагает выполнить три шага для создания проекта и программного файла.

Шаг 1. Определение рабочего пространства (рис. 1.5)

В первой строке всплывающего окна указывается путь рабочего пространства для выбранного МК. Используя кнопку «*Browse*», можно изменить путь рабочего пространства, предложенный программой.

В следующей строке окна предлагается ввести имя рабочего пространства.

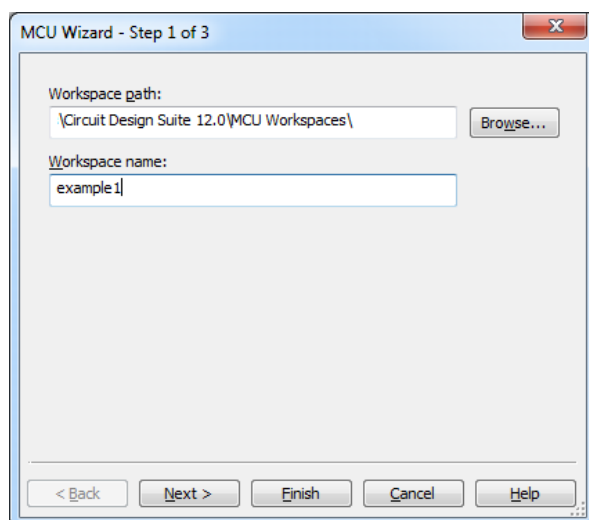


Рис. 1.5. Окно задания рабочего пространства

Шаг 2. Создание проекта для микроконтроллера (рис. 1.6)

В этом окне предлагается установить следующие настройки для будущего проекта:

- 1) тип проекта: Standard;
- 2) язык программирования: Assembly;
- 3) компилятор 8051/8052 Metalink assembler;
- 4) имя проекта.

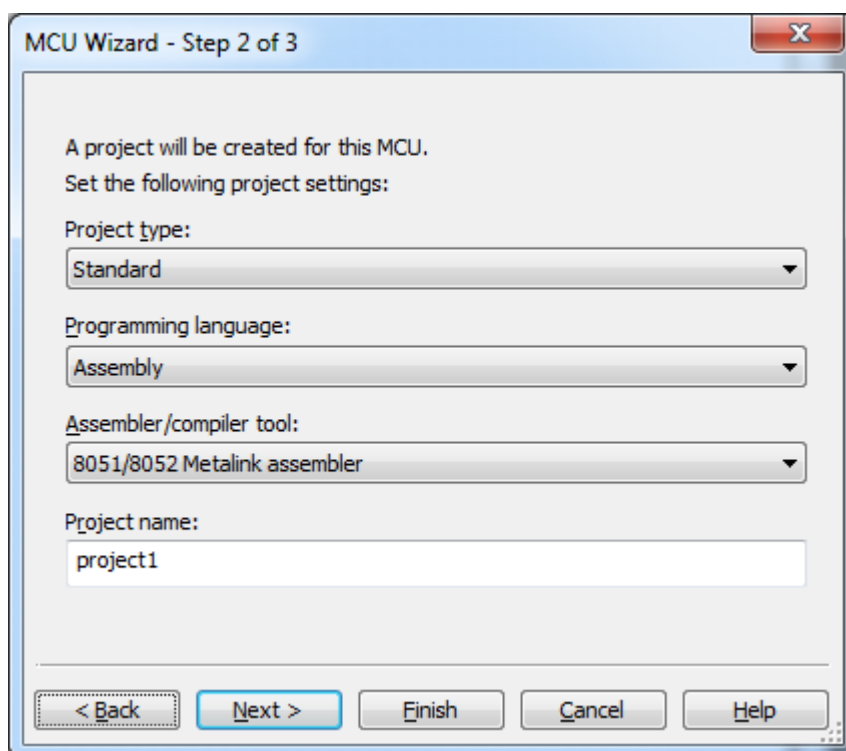


Рис. 1.6. Создание проекта для микроконтроллера

Шаг 3. Создание программного файла (рис. 1.7)

В этом окне предлагается создать либо пустой проект, то есть без программного файла, либо добавить исходный программный файл, указав его имя. Работа с Мастером заканчивается нажатием кнопки Finish.

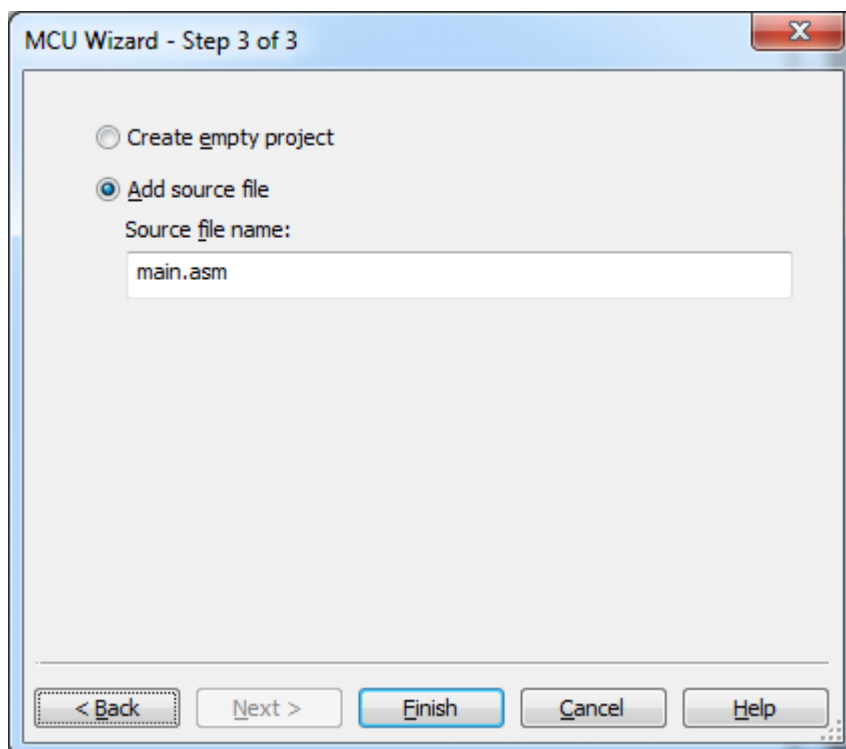


Рис. 1.7. Создание программного файла

После этого будет создан программный файл (рис. 1.8).

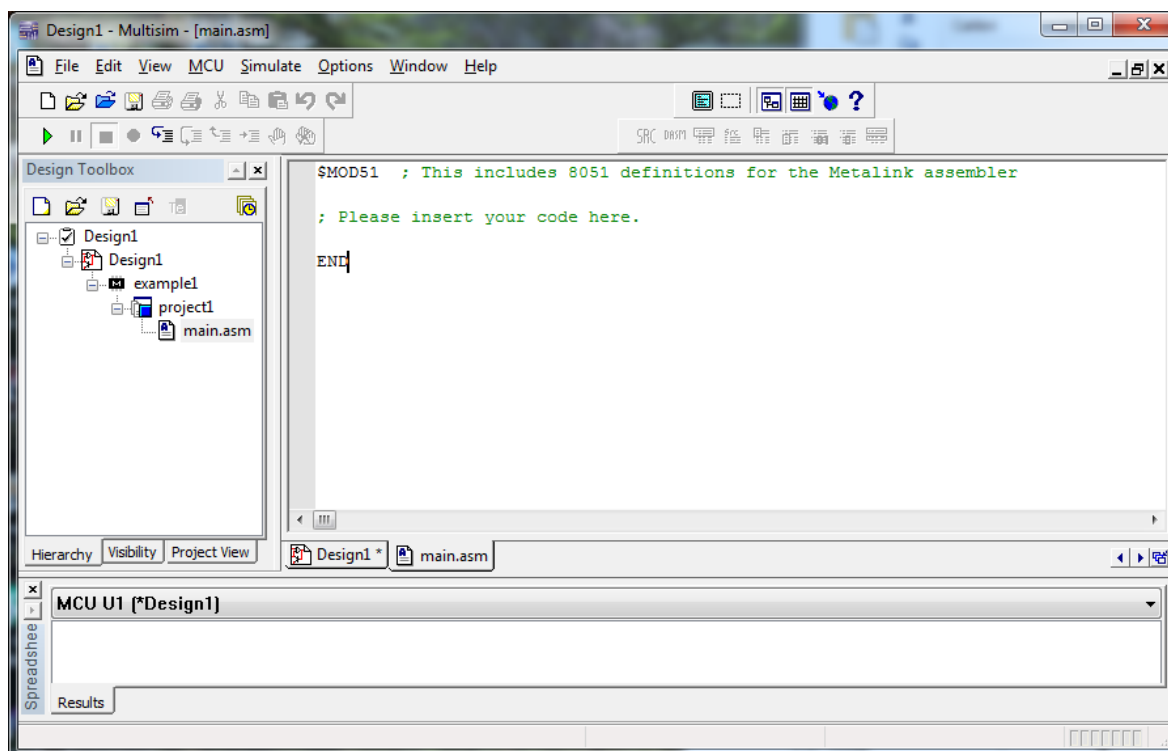


Рис. 1.8. Окно программного файла

Добавьте в программный файл следующий код на языке Ассемблер:

```
$MOD51 ; подключение МК-51
org 00h ;начинаем программу с адреса 00h
start: ;метка начала программы
mov p1,#0h;выключаем индикатор
jnb P2.0, on ;переход на метку on, если с генератора
поступает 1
jmp off ;переход на метку off, если с генератора
поступает 0

on:
setb P1.0 ;включение индикатора
jnb P2.0, on
jmp start ;переход на начало программы

off:
clr P1.0 ;выключение индикатора
jnb P2.0, off
jmp start
END
```

В данной программе при поступлении на вход МК логической 1 загорается индикатор, при поступлении логического 0 индикатор выключается.

Система команд МК 8051 приведена в приложении 1.

Переключитесь на вкладку схемы (Design1) и добавьте в схему индикатор (*Indicators->Probe->Probe_dig_red*), элемент питания(*Sources->Power_Sources->VCC*), генератор (*Sources ->DIGITAL_SOURCES->INTERACTIVE_DIGITAL_CONSTANT*) и землю (*Sources->Power_Sources->DGND*), соедините их с МК. В итоге у вас должна получиться схема, приведенная на рис. 1.9.

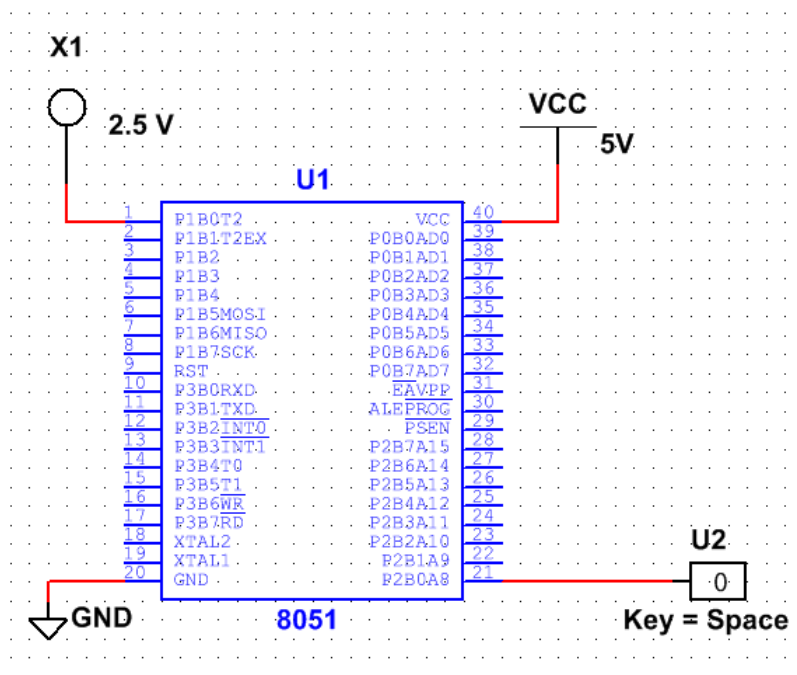


Рис. 1.9. Схема соединения элементов

Всем соединениям присваивается автоматически номер по порядку, начиная с 1. Чтобы изменить нумерацию соединения или присвоить ему логическое имя, необходимо дважды кликнуть по соединительному проводнику.

В данной схеме на один из входов МК подается последовательность 0 и 1 с помощью генератора. Смена 1 и нуля осуществляется по нажатию кнопки *Space*.

2.2. Подключение осциллографа

Разместите на рабочей области виртуальны осциллограф. Символ осциллографа имеет два канала: Channel A и Channel B. Подсоедините «+» канала A к выводу генератора, «-» - к земле. На панели прибора необходимо указать деление временной шкалы, равное 10 мс. Выберите по оси у цену деления для канала A равной 5В. Окно осциллографа открывается двойным нажатием левой кнопки мыши на изображении прибора.

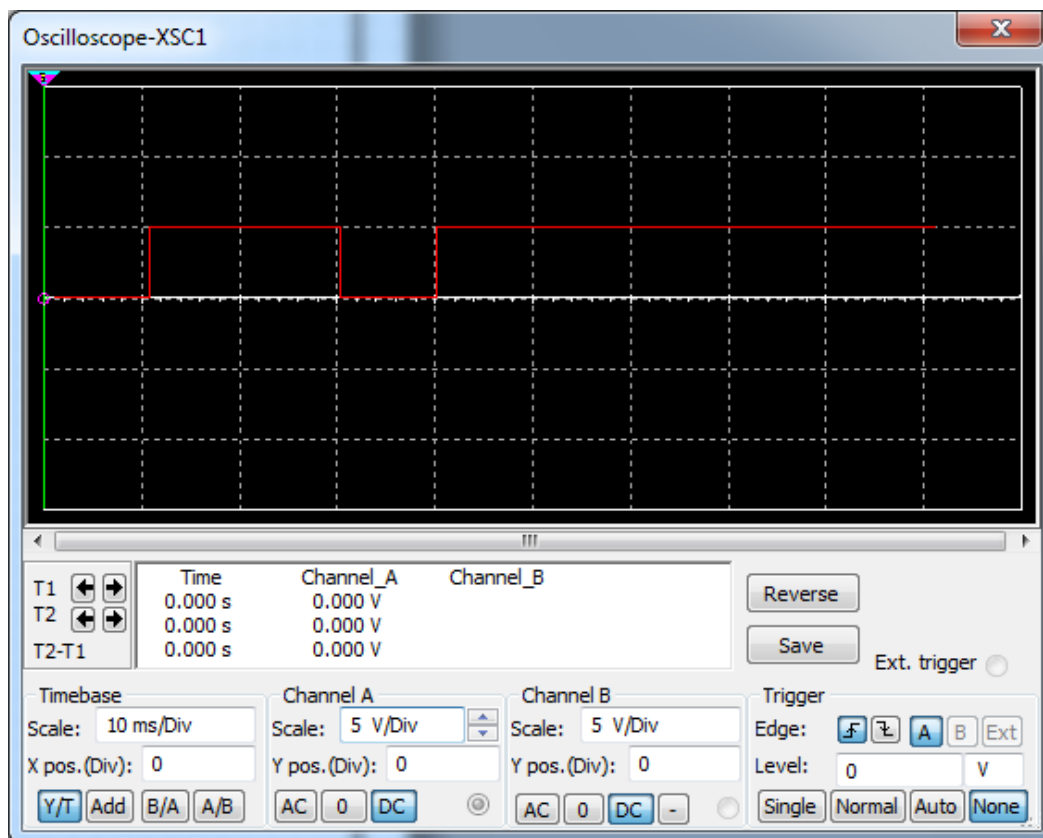


Рис. 1.10. Окно осциллографа

1.3. Моделирование работы схемы

Чтобы откомпилировать и отредактировать программный файл необходимо выбрать пункт *MCU->MCU 8051 UI->Build*. В окне сообщений на экране отражается информация об ошибках и предупреждениях. Если имеются ошибки, в окне сообщений указываются номера строк, в которых они находятся. Для отображения нумерации строк необходимо выбрать меню *MCU->line numbers*.

Моделирование работы схемы производится через меню *Simulate -> Run*, или через соответствующую кнопку на панели Моделирование. При этом открывается окно отладчика. Для пошагового моделирования используется пункт *Step into* в меню *MCU*.

При нажатии кнопки *Pause* моделирование приостанавливается и возможно посмотреть, на каком этапе находится симуляция. Кнопка *Stop* останавливает выполнение моделирования.

Откомпилируйте проект и убедитесь, что в коде нет ошибок. Запустите моделирование и проверьте работу схемы, изменяя поступающий на вход МК сигнал с единицы на ноль с помощью кнопки Space.

Откройте свойства МК и на вкладке Code удалите подключенный программный файл и добавьте новый (рис. 1.11), при этом выберите язык программирования C, и компилятор Hi-Tech C51-Lite compiler (рис. 1.12)

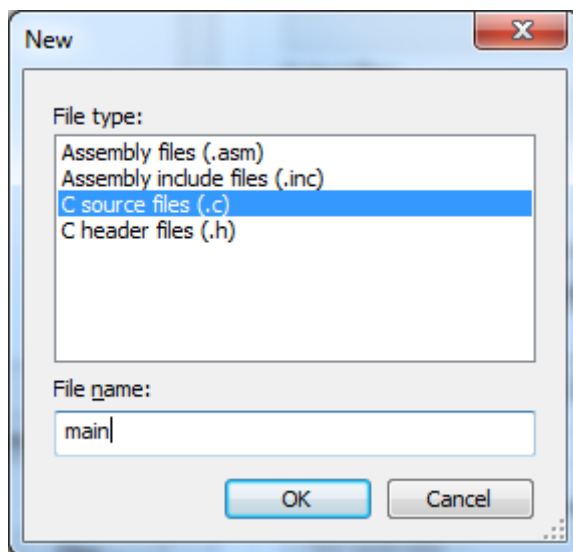


Рис. 1.11. Создание нового программного файла

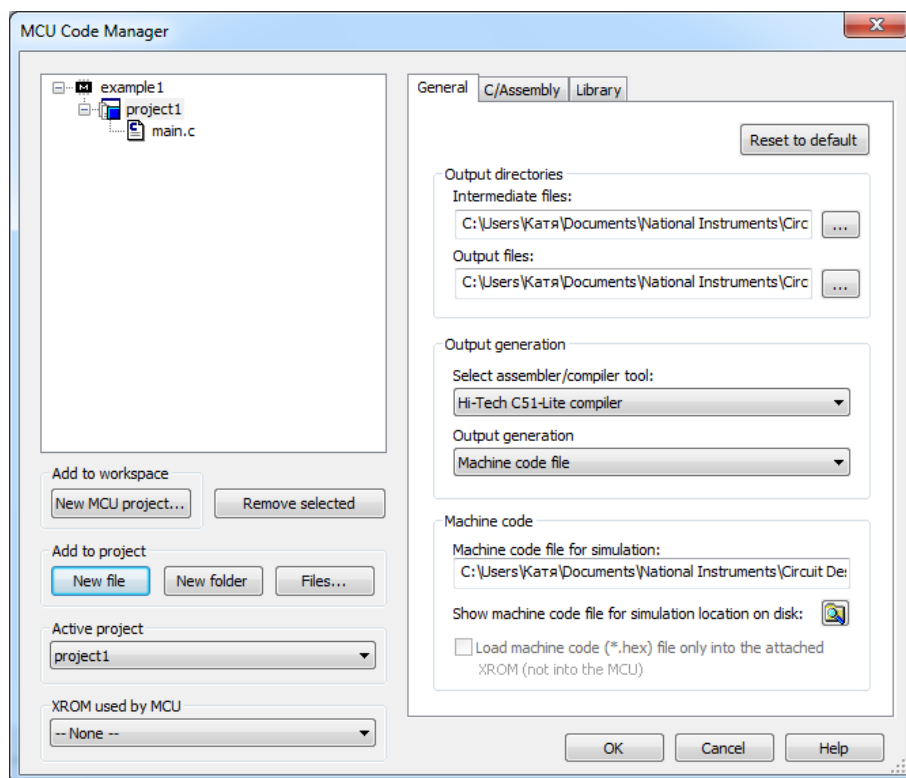


Рис. 1.12. Окно свойств МК

Напишите на языке С программу обработки для МК, выполняющую те же функции, что и в программе на ассемблере. Выполните построение проекта, проверьте правильность работы в режимомоделирования.

Приложение 1
(справочное)
Система команд i8051

Обозначения:

R_n ($n = 0, 1, \dots, 7$) – регистр общего назначения в выбранном банке регистров;

$@R_i$ ($i = 0, 1$) – регистр общего назначения в выбранном банке регистров, используемый в качестве регистра косвенного адреса;

ad – адрес прямоадресуемого байта;

ads – адрес прямо адресуемого байта-источника;

add – адрес прямо адресуемого байта-получателя;

ad11 – 11-разрядный абсолютный адрес перехода;

ad16 – 16-разрядный абсолютный адрес перехода;

rel – относительный адрес перехода;

#d – непосредственный операнд;

#d16 – непосредственный операнд (2 байта);

bit – адрес прямо адресуемого бита;

/bit – инверсия прямо адресуемого бита;

A – аккумулятор;

PC – счетчик команд;

DPTR – регистр указатель данных;

() – содержимое ячейки памяти или регистра.

1. Команды пересылки данных

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор из регистра (n=0÷7)	MOV A, Rn	11101rrr	1	1	1	(A) ← (Rn)
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101	3	2	1	(A) ← (ad)
Пересылка в аккумулятор байта из РПД (i=0,1)	MOV A, @Ri	1110011i	1	1	1	(A) ← ((Ri))
Загрузка в аккумулятор константы	MOV A, #d	01110100	2	2	1	(A) ← #d
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	(Rn) ← (A)
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr	3	2	2	(Rn) ← (ad)
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	(Rn) ← #d
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	(ad) ← (A)
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	(ad) ← (Rn)
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	(add) ← (ads)
Пересылка байта из РПД по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	(ad) ← ((Ri))
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	(ad) ← #d
Пересылка в РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересылка в РПД константы	MOV @Ri, #d	0111011i	2	2	1	((Ri)) ← #d
Загрузка указателя данных	MOV DPTR, #d16	10010000	13	3	2	(DPTR) ← #d16
Пересылка в аккумулятор байта из ПП	MOVC A, @A+DPTR	10010011	1	1	2	← ((A) + (DPTR))
Пересылка в аккумулятор байта из ПП	MOVC A, @A+PC	10000011	1	1	2	(PC) ← (PC)+1, (A) ← ((A)+(PC))
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	1110001i	1	1	2	(A) ← ((Ri))
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	(A) ← ((DPTR))
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	((Ri)) ← (A)
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	((DPTR)) ← (A)
Загрузка в стек	PUSH ad	11000000	3	2	2	(SP) ← (SP) + 1, ((SP)) ← (ad)
Извлечение из стека	POP ad	11010000	3	2	2	(ad) ← (SP), (SP) ← (SP) - 1
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	(A) ↔ (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A _{0...3}) ↔ ((Ri)) _{0...3}

2. Арифметические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сложение аккумулятора с регистром (n=0÷7)	ADD A, Rn	00101rrr	1	1	1	(A) ← (A) + (Rn)
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101	3	2	1	(A) ← (A) + (ad)
Сложение аккумулятора с байтом из РПД (i = 0,1)	ADD A, @Ri	0010011i	1	1	1	(A) ← (A) + ((Ri))
Сложение аккумулятора с константой	ADD A, #d	00100100	2	2	1	(A) ← (A) + #d
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	(A) ← (A) + (Rn) + (C)
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101	3	2	1	(A) ← (A) + (ad) + (C)
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	(A) ← (A) + ((Ri)) + (C)
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100	2	2	1	(A) ← (A) + #d + (C)
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если (A _{0...3}) > 9 или ((AC)=1), то (A _{0...3}) ← (A _{0...3}) + 6, затем если (A _{4...7}) > 9 или ((C)=1), то (A _{4...7}) ← (A _{4...7}) + 6
Вычитание из аккумулятора регистра и заёма	SUBB A, Rn	10011rrr	1	1	1	(A) ← (A) - (C) - (Rn)
Вычитание из аккумулятора прямоадресуемого байта и заёма	SUBB A, ad	10010101	3	2	1	(A) ← (A) - (C) - (ad)
Вычитание из аккумулятора байта РПД и заёма	SUBB A, @Ri	1001011i	1	1	1	(A) ← (A) - (C) - ((Ri))
Вычитание из аккумулятора константы и заёма	SUBB A, d	10010100	2	2	1	(A) ← (A) - (C) - #d
Инкремент аккумулятора	INC A	00000100	1	1	1	(A) ← (A) + 1
Инкремент регистра	INC Rn	00001rrr	1	1	1	(Rn) ← (Rn) + 1
Инкремент прямоадресуемого байта	INC ad	00000101	3	2	1	(ad) ← (ad) + 1
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	((Ri)) ← ((Ri)) + 1
Инкремент указателя данных	INC DPTR	10100011	1	1	2	(DPTR) ← (DPTR) + 1
Декремент аккумулятора	DEC A	00010100	1	1	1	(A) ← (A) - 1
Декремент регистра	DEC Rn	00011rrr	1	1	1	(Rn) ← (Rn) - 1
Декремент прямоадресуемого байта	DEC ad	00010101	3	2	1	(ad) ← (ad) - 1
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	((Ri)) ← ((Ri)) - 1
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	4	(B)(A) ← (A)*(B)
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	4	(B).(A) ← (A)/(B)

3. Логические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \text{ AND } (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \text{ AND } (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \text{ AND } ((Ri))$
Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \text{ AND } \#d$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \text{ AND } (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \text{ AND } \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \text{ OR } (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \text{ OR } (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \text{ OR } ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \text{ OR } \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \text{ OR } (A)$
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \text{ OR } \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \text{ XOR } (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \text{ XOR } (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \text{ XOR } ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \text{ XOR } \#d$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \text{ XOR } (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \text{ XOR } \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow \text{NOT}(A)$
Сдвиг аккумулятора влево циклический	RL A	00100011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div6, (A_0) \leftarrow (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div6 (A_0) \leftarrow (C), (C) \leftarrow (A_7)$
Сдвиг аккумулятора вправо циклический	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div6, (A_7) \leftarrow (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div6 (A_7) \leftarrow (C), (C) \leftarrow (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0...3}) \leftrightarrow (A_{4...7})$

4. Команды операций над битами

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сброс переноса	CLR C	11000011	1	1	1	$(C) \leftarrow 0$
Сброс бита	CLR bit	11000010	4	2	1	$(b) \leftarrow 0$
Установка переноса	SETB C	11010011	1	1	1	$(C) \leftarrow 1$
Установка бита	SETB bit	11010010	4	2	1	$(b) \leftarrow 1$
Инверсия переноса	CPL C	10110011	1	1	1	$(C) \leftarrow \text{NOT}(C)$
Инверсия бита	CPL bit	10110010	4	2	1	$(b) \leftarrow \text{NOT}(b)$
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	$(C) \leftarrow (C) \text{ AND } (b)$
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	$(C) \leftarrow (C) \text{ AND } (\text{NOT}(b))$
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	$(C) \leftarrow (C) \text{ OR } (b)$
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000	4	2	2	$(C) \leftarrow (C) \text{ OR } (\text{NOT}(b))$
Пересылка бита в перенос	MOV C, bit	10100010	4	2	1	$(C) \leftarrow (b)$
Пересылка переноса в бит	MOV bit, C	10010010	4	2	2	$(b) \leftarrow (C)$

5. Команды передачи управления

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Длинный переход в полном объеме ПП	LJMP ad16	00000010	12	3	2	$(PC) \leftarrow ad16$
Абсолютный переход внутри страницы в 2 Кб	AJMP ad11	a ₁₀ a ₉ a ₈ 00001	6	2	2	$(PC) \leftarrow (PC) + 2, (PC_{0-10}) \leftarrow ad11$
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	$(PC) \leftarrow (PC) + 2, (PC) \leftarrow (PC) + rel$
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	$(PC) \leftarrow (A) + (DPTR)$
Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(A)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(A) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен единице	JC rel	01000000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(C)=1$, то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(C)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=1$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит равен нулю	JNB bit, rel	00110000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=1$, то $(b) \leftarrow 0$ и $(PC) \leftarrow (PC) + rel$
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr	5	2	2	$(PC) \leftarrow (PC) + 2, (Rn) \leftarrow (Rn) - 1$, если $(Rn) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Декремент прямоадресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101	8	3	2	$(PC) \leftarrow (PC) + 2, (ad) \leftarrow (ad) - 1$, если $(ad) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101	8	3	2	$(PC) \leftarrow (PC) + 3$, если $(A) \neq (ad)$, то $(PC) \leftarrow (PC) + rel$, если $(A) < (ad)$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	10	3	2	$(PC) \leftarrow (PC) + 3$, если $(A) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $(A) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr	10	3	2	$(PC) \leftarrow (PC) + 3$, если $(Rn) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $(Rn) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение байта в РПД с константой и переход, если не равно	CJNE @Ri, #d, rel	1011011i	10	3	2	$(PC) \leftarrow (PC) + 3$, если $((Ri)) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $((Ri)) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	$(PC) \leftarrow (PC) + 3, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0-7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8-15}), (PC) \leftarrow ad16$
Абсолютный вызов подпрограммы в пределах страницы в 2 Кб	ACALL ad11	a ₁₀ a ₉ a ₈ 10001	6	2	2	$(PC) \leftarrow (PC) + 2, (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{0-7}), (SP) \leftarrow (SP) + 1, ((SP)) \leftarrow (PC_{8-15}), (PC_{0-10}) \leftarrow ad11$
Возврат из подпрограммы	RET	00100010	1	1	2	$(PC_{8-15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0-7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	$(PC_{8-15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1, (PC_{0-7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Пустая операция	NOP	00000000	1	1	1	$(PC) \leftarrow (PC) + 1$