

Chapter 10

Programming in C

Lesson 08

C Compilers for 8051 Programs

C Compiler

A compiler converts the program to object code for a specific processor

ANSI C

- ANSI (American National Standards Institute) C standard
- C99 standard of ANSI C developed in 2000
- Latest standard for C programming
- ANSI C supported by many popular compilers
- A program in ANSI C when not using the hardware specific codes compiles on any CPU or MCU or platform

ANSI C compliant compilers for 8051 Programs in C

- Have the extensions of ANSI C and new library functions for the MCUs
- Provide the additions to the ANSI C functions and data types
- The extensions enable hardware specific compilations

ANSI C compliant compilers

- Take care of the data alignment in memory
- Also take care of the size of the data-types in the specific MCU

Compiler Features

1. Linking files,libraries, assembled and Machine code generation
2. Allocation of appropriate start and end addresses of functions
3. Allocation of addresses for data storage, variables, stacks,queues, tables, messages

C compiler Output

1. Prepares a program Listing, .lst file

2. Prepares list of errors

3. Object code after including the files and library

Error List

Code List

Object code

Compiler Features

4. Macros replaced by statement lines

Data (Values) Assignment

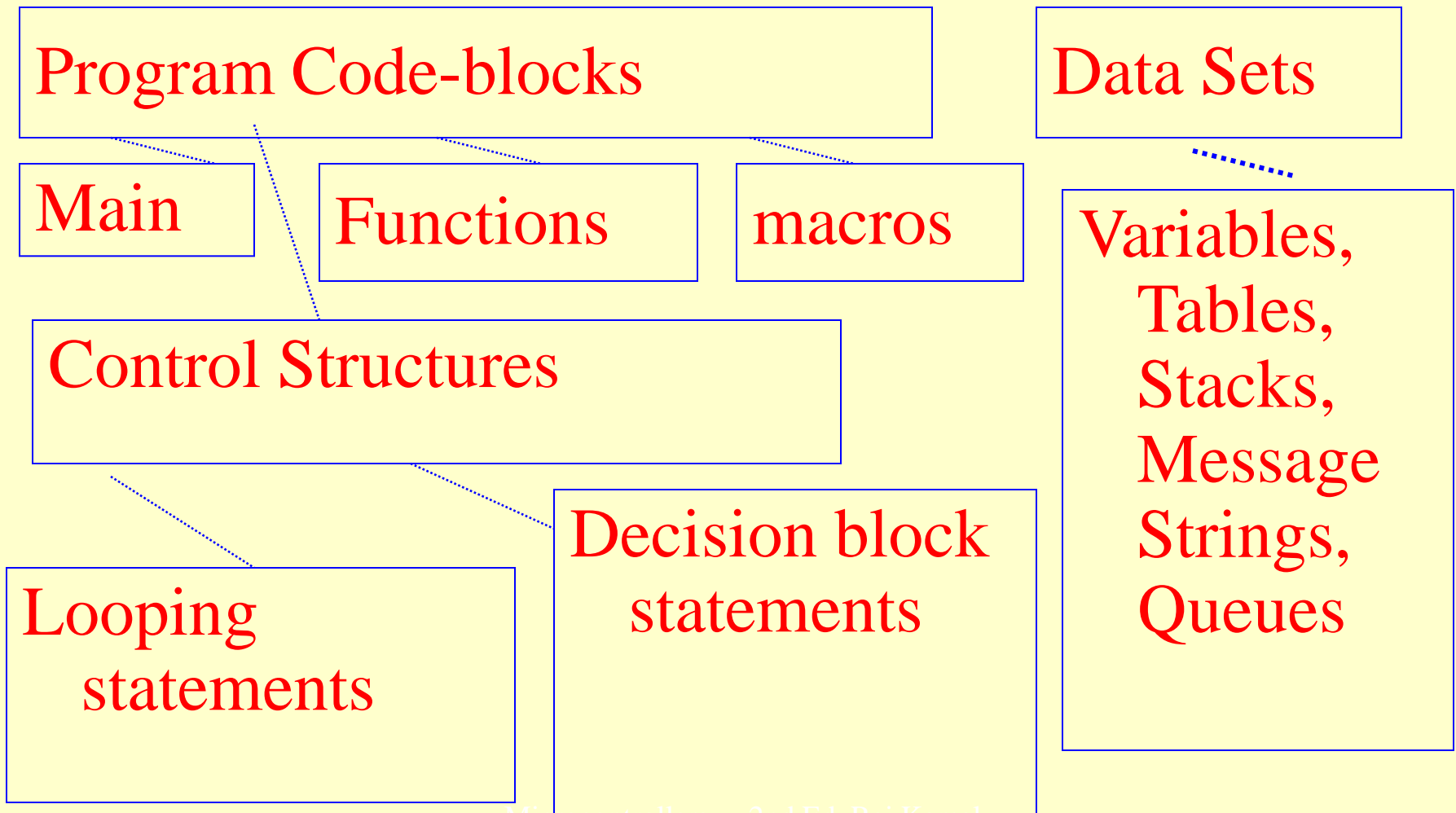
variables

Expressions

Constants

Tables, Stacks, Message Strings, Queues

Compiler Addresses Allocation



Compilers

- GNU
- SDCC
- KEIL

8051 many derivatives of the classic 8051 devices

- The derivatives and IP cores made by Analog Devices
- Atmel,
- Cypress Semiconductor,
- Dallas Semiconductor,
- Goal

8051 many derivatives of the classic 8051 devices

- Hynix,
- Infineon,
- Intel,

8051 many derivatives of the classic 8051 devices

- NXP (founded by Philips),
- OKI,
- Silicon Labs,
- SMSC,
- STMicroelectronics,

8051 many derivatives of the classic 8051 devices

- Synopsis,
- TDK,
- Temic,
- Texas Instruments, and
- Winbond

8051 many derivatives of the classic 8051 devices

- IP core— means an intellectual property files for a specific VLSI circuit or processor or device, which can be put on to the silicon chip using those files at the foundry after integrating with other files.

C compiler for specific derivative or version of MCU support must

- C compiler compiles provided the a specific derivative or version of MCU is supported by that compile

GNU development tools

- Free software (FSF) that can be downloaded from the net
- GNU C/C++ compiler called gcc (GNU compilers collection)

GNU C/C++

- Assembler is called *as*.
- Linker is called *ld*
- GNU linker also provides the scripting
- Script means piece of program code, which is interpreted at run time step by step
- Script enables the control of the object file
- GNU debugger and simulation tools

GNU build system

- Simplifies the development of portable programs
- Enables simple building of the source code programs
- Program developer give more attention to the debugging of the source code
- Installer compiles and installs the program directly from the distribution source code

SDCC (Small Device C Compiler)

- Free open source, optimizing ANSI-C compiler
- The entire source code for the compiler distributed under GPL
- SDCC has latest version SDCC 2.9.4 since December 2009

SDCC

- . It is freely downloadable from website <http://sdcc.sourceforge.net/snap.php>.
- SDCC subversion repository or alternately from the snapshot builds <http://sdcc.sourceforge.net/snap.php>.

SDCC

- Distributed with all files for the projects, workspaces,
- Usable with Visual C++ 6.0/NET (except for SDCDB (SDCC debugger) and ucSim (SDCC simulator))

SDCC user guide

- <http://www.sdcc.sourceforge.net/doc/sdccman.pdf>.

SDCC

- ASXXX a free open source retargetable assembler
- ASLINK linker
- SDCC extensive language extensions suitable for utilizing various microcontrollers
- Underlying hardware effectively used
- Plug-in support for Eclipse RTOS
(<http://sourceforge.net/projects/eclipse-sdcc>).

SDCC features

- Designed for Intel MCS51 based MCUs (8031, 8032, 8051, 8052, etc.), Dallas DS80C390 variants, Freescale (formerly Motorola) HC08 and Zilog Z80 based MCUs
- SDCC can be retargeted
- It can be retargeted for other MCUs
- Support to Microchip PIC, Atmel AVR under development

SDCC features

- Has MCU specific and loop optimisers and has dead code eliminator

Loop optimisation

- Done by loop reversing and reducing strength of the induction variable
- If an induction variable is declared as int data type but does not take int range values but takes only unsigned char values, then it is converted to unsigned char data type. This reduces the strength for optimisation purpose

Dead code eliminator

- The functions, macros and the statements, which are declared but is not used in the source files eliminated

SDCC features

- Has peephole optimizer
- Uses a rule based substitution mechanism
- Presently does not have code banking support in MX derivatives of 8051

Mechanism MCU independent.

- Peephole means peeping into the operation and modifying it into simpler operations
- For example, multiplication can be reduced to shifts and additions.
- $y = 5 \times x$ is reduced to $\text{temp1} = \text{shift } x \text{ left by } 2$ and
- $y = \text{temp1} + x$. [Left shift of x by 2 means multiplication by 4.]

SDCC features

- Has global sub expression elimination,
- Supports seven basic data types, including 32-bit IEEE floating-point and sbit, sfr, sfr16, sfr32.
- Supports flexible variable allocation using bit, data (near), bdata, idata, xdata (far), and pdata memory types.
- Allows inline assembler code

SDCC features

- Has companion source level debugger SDCDB
- Debugger uses ucSim
- ucSim a free open source simulator for 8051 and other MCUs
- Supports constant folding and propagation, copy propagation, jump tables for switch statements.

Constant folding

- Means constants occurring together in an expression are combined to another constant. For example, `#define pi 3.141` and in a function $2 \times \text{pi}$ will be folded to 6.242 at the compilation.

Keil C compiler Cx51

- Keil™ an ARM® company offers the 8051 C compiler
- The Cx51 Compiler allows writing of the 8051 microcontroller applications in C

Executable file efficiency and speed

- Near to the assembly language programs
- Several Language extensions in the Cx51
- Enables full access to all resources of the 8051 or its derivative.

C51 Compiler

- Translates C source files into relocatable object modules
- The modules full symbolic information
- Enables debugging of the object files with the μ Vision Debugger or an in-circuit emulator
- Generates a listing file
- Can be included the symbol table and cross reference information

C51 Compiler

- Supports nine basic data types, including 32-bit IEEE floating-point. There is efficient calculation during float variable operations.

C51 Compiler

- Support flexible variable allocation using bit, data, bdata, idata, xdata, and pdata memory types. Supports to memory allocation and relocation functions, memory freeing functions and user-defined heap sizes. [Heap is the run-time output data in RAM.]
- Enables interrupt functions in C.
- Enables full use of the 8051 register banks.

C51 Compiler

- Has 8051 specific optimiser
- Supports design of the reentrant functions

C51 Compiler

- Support flexible variable allocation using bit, data, bdata, idata, xdata, and pdata memory types. Supports to memory allocation and relocation functions, memory freeing functions and user-defined heap sizes. [Heap is the run-time output data in RAM.]
- Enables interrupt functions in C.
- Enables full use of the 8051 register banks.
- Has 8051 specific optimiser. This optimises the codes
- Supports design of the reentrant functions. It solves the critical section interrupt problems with the shared data among the functions.

C51 Compiler

- Enables interrupt functions in C.
- Enables full use of the 8051 register banks

C51 Compiler

- Enables complete symbol and type information when debugging at the source-level
- Enable use of AJMP and ACALL instructions. [ACALL and AJMP 8051 instructions in which only 11-bits of program counter are specified. The instruction becomes of 2 bytes length in stead of 3.]

C51 Compiler

- Supports bit-addressable data objects,
- Supports constant folding, loop optimisers, dead code eliminator, reduction of strength and peephole optimizer.
- Supports popular emulators
- Has built-in interface for the RTX51 Real-Time Kernel.

C51 Compiler

- uses MCU independent rule based substitution mechanism
- Supports dual data pointers on Atmel, AMD, Cypress, Dallas Semiconductor, Infineon, Philips, and Triscend microcontrollers,
- Supports many 8051 family versions and derivatives, 8051 Extended and 8051 MX Philips 8xC750, 8xC751, and 8xC752 limited instruction sets MCUs

C51 Compiler

- Supports for the Infineon 80C517 arithmetic unit

Summary

We learnt

- GNU tools, SDCC or Keil compilers can be used for programming in C
- Interrupt functions written for the functions, which run on hardware interrupt in 8051
- One can define the register bank for the interrupt function
- Optimiser

End of Lesson 08 on

C Programming Examples for
Ports