Chapter 10

Programming in C

Lesson 02

Memory Constitution, constants, variables and Data Types

8051 Memory Constitution

- Several sections of internal and external memory
- When a data type declared– the memory type can also be declared
- Types of memory can be specified when declaring a variable data type in an 8051compiler

Memory Types

- **code** Memory: Program (code) memory
- Addresses are from 0x0000 to 0xFFFF
- 8051— 64 kB of code memory

Memory Types

- **data** Memory: Directly addressable internal RAM memory for data
- Gives fast access
- 8051—128 byte of data memory

Memory Types

- **idata** Memory: Indirectly addressable internal RAM memory for data
- 8051/52— has full internal space 256 byte of idata memory

Memory Types

- **bdata** Memory: Bit addressable internal RAM memory for data
- 8051—128 bits data memory (16 bytes between 0x20 and 0x2F byte addresses)

Memory Types

- **xdata** Memory: Indirectly addressable external RAM memory for data
- 8051— full internal space 64 kB byte of external data memory

Memory Types

- **pdata** Memory: Paged external data memory of 256 bytes
- Accessed by simple instruction MOVX @R1 or MOVX @R2.

Memory Types

- **far** Memory: Extended RAM memory spaces after 64 kB
- 8051 versions, for example Philips 80C51MX— 8MB far memory
- 8 MB pointed by 3-byte address, depends on compiler
- Accessed by user certain defined functions for the specific 8051 versions

Memory Types

- **const far:** Extended memory spaces after 64 kB
- 8051 versions, for example Philips 80C51MX— 8 MB far memory
- Pointed by 3-byte address or 24-bit DPTR register, depends on compiler
- Accessed by user certain defined functions for the specific 8051 versions

Constant

- A constant one, which is assigned a fixed value
- Once assigned a value, it remains fixed all through out the program
- Constant saved in RAM, if it does not change from within a program

Constant

- Advised to give names in capital letters for the constants
- Constant NUMLEDS— number of LEDs in the system
- NUMLEDS saved in RAM

define NUMLEDS 4 /* Four number LEDs in the system*/

Constant

- Saved in program memory ROM flash, if it does not change from functions in one program to functions in another program
- For example, memory size of flash— constant for given hardware
- Addresses of ports also constant
- Port addresses also same for one function to function

Variable

- A variable is one which is assigned a value, which can change
- Saved at a memory address (internal or external) or register or special function register

Variable Example

- P1 = 0x10 means P1 variable at an address; assigned hexadecimal value = 0x10
- When P1 corresponds to Port P1 of 8051
- Hence statement, P1 = 0x93 when executed makes P1 port 8 bits = 10010011 at the SFR address 0x90

Variable

- May be explicitly assigned to a specific memory space (by including a memory type specified in the declaration) or implicitly assigned (based on the memory model)

Variable portdata

- char data portdata; /* portdata is a variable, it is explicitly assigned as of data memory type and variable data type is of a character. */
- data char portdata.; /* This is also equivalent to above acceptable in certain old style compiler version.*/
- char portdata; /* portdata is a variable, it is implicitly assigned as of data memory-type and variable data-type is of a character if memory model in the function is small memory model.*/

Assignment of value to a variable

- char data portdata = 0x7D; /* portdata is a variable, it is explicitly assigned as of data memory type and variable data type is of a character. The variable address has the 8-bits for 0x7D. */
- data char portdata. = 0x7D; /* This is also equivalent to above acceptable in certain old style compiler version.*/
- char portdata= 0x7D; /* portdata is a variable, it is implicitly assigned as of data memory-type and variable data-type is of a character if memory model in the program is small memory model. The variable address is in internal RAM and has the 8-bits for 0x7D */

Two types of variables

- Local
- Global
- Local variables usable only within a function (routine)
- Global variables used and changeable in two or more functions

Global variable in preprocessor directive

define long time_date /* It means *time_date* is a global variable of data-type as long. The long is used in SDCC as 32-bit number*/

Global variable in preprocessor directive

define bool *start_switch* /* It means *start_switch* is a global and data type of it is Boolean variable. Boolean variable is one bit variable and it can take two values–true or false (1 or 0)

/The bool data type provided in SDCC (small device C compiler). */

Data Type

- A variable or constant data can be one of the data types permissible by compiler

C compiler allowed data types

- *char* (8-bit) for ASCII character
- *char [ ]* an array of characters for a string
- A name is a string of characters
- For example, Dr. Raj Kamal is a string of 13 characters (space and full stop signs)
- Above String takes14 bytes as last character is null

C compiler allowed data types

- *byte* (8-bit)
- *float* (32-bit)
- *double* (64-bit double precision IEEE754 standard)

C compiler allowed data types

- *unsigned short* (16-bit)
- *short* (16-bit)
- *unsigned int* (32-bit)

C compiler allowed data types

- *int* (32-bit)
- *long* (64-bit)

Small Device C compiler (SDCC) data types

- *bool* (1-bit)
- *char* (8-bit) for ASCII character
- *char [ ]* an array of characters for a string

Small Device C compiler (SDCC) data types

- *float* (32-bit, IEEE754 standard)
- *unsigned short* (16-bit)
- *short* (16-bit)
- *unsigned int* (16-bit)
- *int* (16-bit)
- *long* (32-bit).

sfr

- sfr P1 0x90; /* P1 is an SFR and the address of SFR is 0x90. */
- sfr P3 = 0xB0;   /* Port P3, address 0B0h */

sfr16

- sfr16 T2 0xCC; /* T2 is an SFR of 16-bit, the lower byte TL2 is at 0xCC and higher byte at 0xCD. */

sbit

- sbit TR0 = TCON^4; /* TR0 is SFR bit. It is b4 bit in TCON.*/
- sbit TR0 = 0x88^4;/*TR0is SFR bit, b4 bit at SFR address 0x88. Note: TCON has address 0x88. */
- sbit TR0 = 0x8C; ;/*TR0 is SFR bit at bit address 0x8C */

bit

- bit isGlowing = 0; /* The variable isGlowing is of 1-bit and at address of isGlowing, the value = 0 is assigned. */

bit

- static bit isComplete = 1; /* The variable isComplete is of 1-bit and at address of isComplete, the value = 0 is assigned. */

bit

- bit data isGlowing = 0; /* It also specifies that bit memory type is data. */

bit

- bit idata isGlowing = 0 ; /* It also specifies that bit memory type is idata. */

register

- register data r7 = 0xB0; /* The register r7 is a register. It is assigned a value = 0xB0.*/
- register data TCON = 0x10; /* The register TCON is assigned a value = 0x10.*/

Memory Models in 8051

- Compact
- Small
- Large

Classic 8051 versions Stack

- Provide for stack the accesses to the internal data memory
- Limited to 256 bytes
- The stack is used how, it is compiler dependent. The Keil Cx51 Compiler locates the stack area immediately following all variables in the internal data memory. The stack pointer accesses the internal memory indirectly.

Compiler dependent stack use

- Keil Cx51 Compiler locates the stack area immediately following all variables in the internal data memory
- The stack pointer accesses the internal memory indirectly

Summary

We learnt

- An 8051 compiler allows declaration of memory types
- Memory types can be data, xdata, idata, pdata, code, far
- Variables and data types

We learnt

- char, char [], float, short, int, long and bool data types
- The operations done on the constants and variables
- Memory models

End of Lesson 02 on

Memory Constitution, constants, variables and Data Types