80C188XL
DEVELOPMENT TOOLS

EE3232 DIGITAL SYSTEMS III
CLASS NOTES CHAPTER 3

Department of Electrical Engineering
University of New Brunswick

 January 5, 2000

SUMMARY

1. Development tools.
- The compilation process.
- Relocatable program modules and location.
- SBC188 compiling steps.
- SBC188 assembly steps.
- Debugging tools.
- Debugging logical errors.
- Debug commands.

DEVELOPMENT TOOLS

Editor : Create/edit .C and .ASM source code as a text file.

Assembler : Accepts a .ASM source file, and translates it into a .OBJ object file and a .LST listing file that lists machine code, values assigned to symbols and syntax errors.

Compiler : Accepts source file(s), and translates it into an .OBJ object file and a (optional) .ASM file and identifies syntax errors.

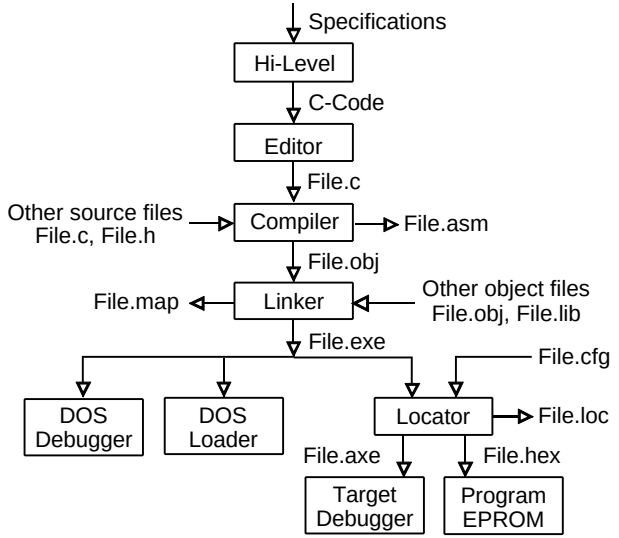
Linker : Accepts one / more .OBJ object files and .LIB library files creating an .EXE executable file. and a .MAP file containing address info on symbols..

Locator : Assigns absolute addresses for loading executable code.

Loader : Loads the code into memory and starts execution.

Debugger : Loads code into memory and allows debugging: single step, examine/modify registers, memory and IO ... etc.

THE COMPILATION PROCESS



RELOCATABLE PROGRAM MODULES AND LOCATION

- Programs are said to be ***relocateable*** if the address(es) of where the program code is to reside in memory is specified by the loader or locator (rather than the assembler or compiler).
- This is a useful feature for an operating system that has to load and run executable programs, (like DOS, Windows, QNX, LINUX ...)
- In embedded applications (that do NOT use an operating system) the program code resides in EPROM at absolute or fixed addresses.
- An absolute address must be assigned to the program code before the EPROM is burned in an embedded application!
- We will use Paradigm LOCATE to convert a .EXE relocateable program into a non-relocatable .AXE program that may be loaded into the SBC188 and executed!

SBC188 COMPILING STEPS

- Create a “C” source program ***filename.c*** .
- From a DOS prompt type ***mc filename.c*** .
- mc is a batch command file that :
  - Compiles the source program using ***bcc***.
  - Links the object file using ***tlink***.
  - Locates the executable using **locate**.

SBC188 ASSEMBLY STEPS

- Create a “ASM” source program ***filename.asm*** .
- From a DOS prompt type ***ma filename.asm*** .
- ma is a batch command file that :
  - Assembles the source program using ***tasm***.
  - Links the object file using ***tlink***.
  - Locates the executable using **locate**.

DEBUGGING TOOLS

- Turbo debug, ***TD***, debugs DOS type applications.
- A low level DOS debugger is ***debug***.
- To debug on the SBC188 remote system : PDRT186 runs on a DOS machine and communicates with a low level ***monitor*** program that runs on the target system for debugging.

DEBUGGING LOGICAL ERRORS

1. Ensure that the program algorithm is correct.
2. Develop the program in modules. This simplifies the debugging process.
3. Before running your program ensure that all IO devices and cables are in working order. Use visual inspection and the monitor Port IO commands to exercise the IO devices.
4. Debug the program (using TD, PDRT186, DEBUG). Run or single step the program examining registers, flags and memory variables to validate program operation.
5. For larger programs set breakpoints at strategic locations. If the program does not execute properly then go to Step 1.

DEBUG Commands

- Debug runs on a PC. Type ***debug filename*** under a DOS prompt.

>H	Help.
>R	Display registers
>R AX	Display AX and prompt for a new value.
>R F	Display Flags.
>D	Display memory beginning at DS:IP
>D DS:100	Display memory beginning at DS:100
>N 16	Change radix to base 16 (default is decimal).
>EB CS:100	Enter bytes (separated by spaces) into memory beginning at CS:100.
>U	Unassemble.
>T 8	Trace 8 instruction cycles beginning at CS:IP.
>BP addr	Set breakpoints
>BC	Clear breakpoints.
>Q	Quit.