

DATE: 25 April 1989

## 1.0 SOFTWARE PRODUCT DESCRIPTION

Product Name: MOLE-HPC-IBM-CR  
Product Number: 970420070-005  
Revision: Betasite C4.c  
System: MS-DOS 2.0 or later (5-1/4" PC-DOS Format)

Overview: This package contains software for compiling programs written in the C language for National Semiconductor's HPC series of micro-controllers. The software executes on an IBM PC, PC/XT, PC/AT, or equivalent clone running MS-DOS or PC-DOS.

This package contains the following program, and additional utilities and support files:

Name	Rev	Date	Description
CCHPC	C4.c	25 Apr 89	C Compiler for HPC

## 2.0 OPERATING ENVIRONMENT

This package runs under the MS-DOS (or PC-DOS) operating system, Rev 2.0 or later, on an IBM PC/XT/AT or compatible. The system must have:

1. At least 400 Kbytes of free memory.
2. A 5-1/4" floppy disk drive, capable of reading Double Sided Double Density PC-DOS 2.x format diskettes.
3. An RS232-compatible asynchronous serial communications port if used with a MOLE development system.

All programs use standard MS-DOS 2.x function calls.

---

MS-DOS is a registered trademark of Microsoft Corp.  
PC-DOS, IBM PC/XT/AT are registered trademarks of IBM.  
MOLE and HPC are registered trademarks of National Semiconductor Corp.

### 3.0 PACKAGE CONTENTS

#### 3.1 OVERALL CONTENTS

Item	Part Number	Description
1	424410883-001	Manual - Title: 'HPC C Compiler User's Manual'
2	440420070-005	5-1/4" Floppy Diskette - Label: 'MOLE-HPC-IBM-CR Betasite C4.c 25 Apr.1989 HPC C Compiler (Disk 1 of 2)'
3	440420070-015	5-1/4" Floppy Diskette - Label: 'MOLE-HPC-IBM-CR Betasite C4.c 25 Apr.1989 Library Source Disk (Disk 2 of 2)'

#### 3.2 CONTENTS OF ITEM 2

This floppy diskette contains the following files:

File Name	File description
CPP.EXE	C Preprocessor, invoked by CCHPC
CCHPC.EXE	C Compiler for HPC
LIBRARY.LIB	C RunTime Library
LIBR_FP.LIB	C RunTime Library with Floating Point
CRTFIRST.ASM	User tunable source for CRTFIRST.OBJ
CRTFIRST.INC	Standard source for CRTFIRST.OBJ
FLOAT.H	Header file for user programs
LIMITS.H	Header file for user programs
STDARG.H	Header file for user programs
STDIO.H	Header file for user programs
SAMLIBR.LIB	Additional library for samples
SAMFIRST.ASM	Tuned source for samples' CRTFIRST
SAMPLE1.C	Sample program
SAMPLE2.C	Sample program
NOTHPC.H	Header file for samples not on HPC
COMPILE.BAT	Compile, Assemble and Link sample programs
XCRC.EXE	CRC calculation program
HPC16083.LIB	HPC Register definition library
CREL.LET	Copy of this Betasite letter
\XCRC\OUT	CRC output file

### 3.3 CONTENTS OF ITEM 3

This floppy diskette contains the following files:

File Name	File description
MAKE-LIB.BAT	Batch file to create libraries
CRTFIRST.ASM	User tunable source of CRTFIRST
CRTFIRST.INC	Standard source of CRTFIRST
CRTINIT.ASM	Source of CRTINIT
BANKSWCH.ASM	Source of BANKSWCH for bankswitching use
LIBRARY.ASM	Source of LIBRARY comments, etc.
LIBR*.ASM	RunTime operations of LIBRARY
LIBI*.ASM	Integer operations of LIBRARY
LIBF*.ASM	Floating operations of LIBRARY
LIBFTRP.ASM	User tunable source for FP Error Trap
LIB*.C	C functions, including printf()
PUTCHAR.C	Memory buffer version of putchar()
SAMFIRST.ASM	Tuned source of CRTFIRST for samples
HPC16083.INC	Include file for HPC register definition
HPC16083.ASM	Source for HPC register definition
XCRC.EXE	CRC calculation program
\XCRC\OUT	CRC output file for Library Sources

#### 4.0 CHANGES FROM PREVIOUS REVISIONS

##### 4.1 REVISION C4:

The following changes have been made:

- a. Library now contains support for bankswitching (see MOLE-HPC-IBM-R betasite letter for linker changes). The library is set up to use PORTB (upper byte) as bank switch port. CRTINIT initializes port bits 8,9,13,14 as output. BANKSWCH contains bankswitch routine. If another bank switch port address or bits are used, the .ASM files should be altered, and library modules replaced. The only changes necessary are two lines in CRTINIT.ASM which initialize LD BFUN,#0 and LD DIRB,#0x6300, and the one line in BANKSWCH.ASM which defines bank\_switch\_port=0x00e3:b.
- b. All initialization sections generated by CCHPC are now ROM8 sections.
- c. User functions in library now contain ROM8 code sections.
- d. Linker requires that all sections fall into ranges now (see linker manual). BASE must be defined to include at least range 0:1 (as this location is reserved). ROM16 (or ROM8) must include range of vector table, typically 0xFFD0:0xFFFF.

#### 4.2 REVISION C3 (BetaSite):

The following bugs have been fixed:

- a. Conditional expressions involving bitwise operations with characters would generate incorrect results in some cases because the character operands were not promoted to integer.
- b. Use of unsigned BASEPAGE characters as index to an array would generate incorrect code rejected by the assembler.
- c. Assignment to the bit fields of structure of byte or less causes the compiler to generate code which result in a warning from the assembler (eg. byte operand used with 16 bit immediate value)
- d. The Compiler was generating incorrect code for expressions of the form P-I and P=I where P is a pointer type and I is an integral type.
- e. The Compiler was not diagnosing an error when a pointer was subtracted from an integer.
- f. The Compiler generated warning for old fashion constructs even though the old\_fashion switch is specified. Warnings are removed.
- g. The Compiler accepted assigning structures of different types.

- h. When two successive pointer operations were made on two different pointers, the compiler generated incorrect code for the second operation.
- i. The Compiler generated an assembly source file even though fatal error(s) had occurred.
- j. The Compiler in some cases generated incorrect code on performing '+' and '-' operations where operands were bit field of structures.

#### 4.3 REVISION C2:

The following bugs have been fixed:

- a. Assignment of a variable to the bit field(s) of a structure of byte long or less caused the illegal operand 'A.L' to be generated.
- b. In some cases, the Compiler with the 8bit mode flag set generated duplicate labels for the switch statement.
- c. NOLOCAL functions sometimes caused the compiler to generate an undefined symbol.
- d. Assignment of a variable of type unsigned char to the bit field of a structure which is located in the upper byte of a word caused the compiler to generate an error message.
- e. The compiler mistakenly removed as redundant, the Accumulator load instruction which was preceded by a single bit operation that altered the value to be loaded.
- f. Multiple structure assignment where the rightmost structure is returned by a function caused the compiler to adjust the stack pointer incorrectly:
- g. The optimizer did not recognize that operations of the form var <OP> = CONST (where var is a static or external variable and <OP> is one of + - & † operators) was affecting the state of the variable which was believed to have valid copies of the accumulator.
- h. In some cases, the Compiler was assuming that the value of a variable was in the Accumulator when in fact the value had been changed by an Increment or Decrement operation. The Compiler failed to reload the changed value.
- i. The Compiler was generating incorrect code when a variable was shifted right by a constant greater than 4 and the variable was a global static or external.

- j. When two pointer register variables were declared in a block and the block contained complex arithmetic operations on arrays of structures containing bit fields, The Compiler generated code using K register to hold the array index and the structure mask value in a conflicting manner.
- k. Expressions of the form,  $V1 = V2 \langle OP \rangle V3$ , where  $Vx$ 's are variable names and  $V2$  and  $V3$  are static or external long variables and  $\langle OP \rangle$  one of '+', '-', '&', '|' or '^' caused the compiler to generate wrong code.
- l. The compiler does not complain about the function definition when the arguments end with a ',' eg. `func(a,b,)`.
- m. The compiler generates a syntax error when a comment is placed on the same line as the include statement.

#### 4.4 REVISION C1:

The default extension of the C source file is '.C' and the option file is '.CMD'.

Command line does not require white space characters for option delimiters.

An additional option, `CHiprev=revision`, is added with revisions of B or C allowed. `CHiprev=B` (default) will overcome the `MULT`, `DIV` and `DIVD` instruction bugs if the destination is an off-chip direct address. The `CHiprev=C` option must be used when the bug is fixed in Rev.C of HPC Core.

In certain cases where the filename contained the drive name, the Compiler generated illegal section names. This bug has been fixed. Can now use member of structure returned by a function. Volatile data type modifier is treated correctly. Prototyping and Trigraphs are supported according to ANSI.

A problem with `BASEPAGE` extern generating illegal assembler statements has been fixed.

Code generation has been significantly improved in many cases, with the emphasis on small code. Data sizes also have been reduced in some cases.

Register variables are now supported. Only variables of type pointer will be put in registers. Only two registers are used; any more than two register declarations are ignored.

Aggregates (structures, unions, arrays) composed only of type char are now considered to be of type char themselves. Thus, their size and alignment is now byte, and can be placed in 8-bit memory. Previously, all aggregates were of type int, with size and alignment of word.

The libraries of support routines currently are true library modules, and the linker extracts as needed.

Argument declarations now accept the declaration of structures and function prototypes within them.

#### 4.5 REVISION C0:

The libraries of support routines currently are single object modules, and the entire module is linked in rather than the few routines being extracted.

##### C-series revisions (from B-series revisions):

Beginning with Revision C0, the C Compiler is based on the Draft ANSI Standard for the C language (X3J11/86-157), no longer on the Kernighan and Ritchie version. Floating point numbers and the operations thereon are supported. The keywords void and enum, structure assignment, structure arguments, and structures return from functions were already supported. The additional keywords const, volatile, and signed, and the function prototyping with argument checking are fully supported.

The compiler now supports separate compilation of modules and the use of externals, supported by a relocating assembler and linker. Globals, static globals, static functions, and externs, extern globals, extern functions are supported.

The linkage and memory placement are now handled by a separate linker program, so the memory and section commands MEMORY, CSECT, DSECT, etc., have been eliminated.

The special keywords BASEPAGE, NOLOCAL, INTERRUPTn, ACTIVE, loop, and switchf have all been retained. The keyword CONST has been eliminated in favor of the standard const keyword. Embedded assembly code is still supported.

Routines needed by the compiled code are provided in a library in object code form. Additionally, a limited library of C routines for the user is available in object form as a sample library for the sample programs.

The source diskette contains source code for all the library code provided.

## 5.0 COMPATIBILITY

### 5.1 COMPATIBILITY WITH PREVIOUS REVISIONS

Revisions from earlier C-series:  
See section 4.0 for changes.

Revisions from B-series:

The changes are largely upward compatible. Memory and section commands must be eliminated, and CONST must be changed to const (possibly with a #define). Anachronisms such as =+ are no longer supported, unless the Old-Fashioned switch is used in invoking compiler. Any embedded assembly code must be rewritten. Operation of the compiler, both in its invocation and interaction with the assembler and linker, have changed significantly. No commonality should be assumed in operation.

### 5.2 COMPATIBILITY WITH RELATED PRODUCTS

#### 5.2.1 HPC MicroControllers

This software is compatible with HPC 16083 Rev B and later chips in the HPC family.

#### 5.2.2 MOLE-HPC-IBMR (part number 970420071-005 Rev. E5)

CCHPC generates source code compatible with ASMHPC Rev. E5, as contained in MOLE-HPC-IBMR Rev E5. Earlier revisions of the ASMHPC program are not guaranteed to work with this revision of CCHPC.

## 6.0 INSTALLATION PROCEDURE

Run the program XCRC \*.\* on the root directory of the HPC C Compiler's diskette. This will get a CRC value for each program and the sum of all the programs. Then compare the results with the file \xcrc\out on the diskette. The resulting CRC's should be the same. Repeat the above procedure for verification of the Library Sources' diskette.

Copy all files on the HPC C Compiler diskette into directory \HPC on your system disk. Include that directory in the search path using the DOS command "PATH". If necessary, edit the file CONFIG.SYS in your root directory so that "FILES=16" or greater; this allows CCHPC to open all its necessary files.

The source code files on the diskette can be put anywhere, but keeping them with the other files will be convenient.

If the .H files are put in another directory, the environment variable "CCHPC" (to affect only the compiler) or "HPC" (to affect the full set of programs) can be set to that directory string and the compiler will find them (see SET in the DOS manual).

The second diskette containing the source code for the library is only necessary if changes are envisioned to other than the CRTFIRST module. The files may be put anywhere, and the objects created moved to \HPC or wherever the linker(LNHPC) will be directed to look for them.

## 7.0 VERIFICATION PROCEDURE

### 7.1 SAMPLE COMPILATION, ASSEMBLY AND LINKAGE

Verification can be accomplished by using COMPILE.BAT to compile, assemble, and link the sample files SAMPLE1 and SAMPLE2. Type COMPILE SAMPLEx (where x is 1 or 2) to generate a .COF executable file, which can be downloaded to MOLE using COMHPC (in package MOLE-HPC-IBM-R). The programs can then be executed; they use printf to write messages to a MOLE memory buffer at 0xE002. The batch file COMPILE.BAT should be of some interest as guidance to linkage of compiled programs and linkage order. Copying these programs to another directory and compiling them there, without moving any files there other than the two files, will confirm that all files are available and that the environment is correctly set.

## 8.0 KNOWN SOFTWARE BUGS

- a) The Assembler/Linker/Librarian package limits filename for the Compiler to the following characters only:
  - Alphanumeric (Letter or digit)
  - Dollarsign (\$)
  - Underbar (\_)This is because the C compiler generates a section name containing the filename; the assembler restricts valid symbol characters in the section name.
- b) If the output of the Compiler is redirected to a file or device, the warning and error messages from the PreProcessor are not redirected.
- c) A "/\*" in the middle of a comment generates a warning about nested comments.
- d) The frame pointer location word location 0xBE is NOT reserved by library; DO NOT include locations 0xBE:0xBF in BASE ranges for linker.

## 9.0 GENERAL CONSIDERATIONS

### 9.1 The Compiler places extra comments in the assembly file generated:

```
; member b1 at offset 0 (7 bits at bit 1)
; member b8 at offset 0 (7 bits at bit 8)
; xr0 size 2
; xr1 size 2
; ir0 at stkloc 2 (0x2) size 2
; ir1 at stkloc 4 (0x4) size 2
```

These comments give information about offset of structure members within the structure, size of structures and offset of local variables with respect to the frame pointer.

## 10.0 RELATED DOCUMENTATION

1. DISK OPERATING SYSTEM (DOS) manual (published by IBM)
2. Draft American National Standard for Information Systems - Programming Language C (ANSI Document Number X3J11/86-157)
3. A C Reference Manual, Harbison & Steele (Prentice-Hall)
4. HPC Assembler/Linker/Librarian User's Manual (Order Number 424410836-001).
5. HPC Communication Program (COMHPC) User's Manual (Order Number 424420188-001)
6. MOLE Brain Board User's Manual (Order Number 420408188-001)
7. HPC Personality Board User's Manual (Order Number 420410447-001)
8. HPC Development Board User's Manual (Order Number 420410901-001)
9. HPC16083/HPC16043/HPC16003 User's Manual (Order Number 424410897-001)
10. Datasheet - HPC series of microcontrollers

## 11.0 USER SUPPORT

### 11.1 Technical support numbers are:

- a. (800) 672-1811 California only.
- b. (800) 538-1866 Continental U.S.
- c. (800) 223-3248 Canada only.
- d. (408) 749-7470 Outside U.S. and Canada

### 11.2 Other numbers are

- a. (408) 739-1162 is Dial-a-Helper, a 24 hour electronic bulletin board information system provided by Microcontroller Applications. A 300 or 1200 baud modem and a terminal or host with terminal emulation capability are required.
- b. (408) 721-5582 is the Microcontrollers Applications number for questions relating to other than HPC software.