124-12

TO: DISTRIBUTION

SUBJECT: C CROSS COMPILER FOR AQUARIUS PROGRAM DEVELOPMENT

The Vandata Z80 C cross compiler on the VAX is now available for general use. It works in conjuction with the Nuvatec Z80 cross assembler (Z80as), so that C and assembler can easily be mixed. "cc80" is the UNIX command used to invoke the compiler, and by typing "man cc80", the cc80 UNIX manual pages are printed at the terminal.

Two advantages have come from this work. The flexible Nuvatec linker is used which allows for user control of the start address. This feature is necessary to generate ROMs for software that exceeds 8k bytes in size. The second advantage is that standard Z80 instruction mnemonics are used by z80as. This will allow for easier use of existing Z80 software.

The following pages describe the use of cc80 in more detail.

Programmer: Chan Min, Bill Taylor

WPT:gt

DISTRIBUTION

Hugh Barnes
Gabriel Baum
Ron Carlson
Dave Chandler
Don Daglow
Richard Decker
Bill Fisher
Russ Haft
Jim Haupt
Bob Hogue
Les Hutchinson
Don Mills
Mike Minkoff
Tom O'Brien
Larry Pumphrey
Keith Robinson
Chuck Rudd
Joey Silvian
Ron Surratt
Rick Timmins
Mark Urbaniec
Mike Winas
Systems Software Group

INTRODUCTION

Cc80 is the combination of the Vandata C compiler and the Nuvatec assembler/loader. The loader is invoked only when a certain flag is found in the argument list. However, a successful completion of loading will always invoke the Intel hex conversion program thus creating 'a.hex' by default, which may be downloaded into a Z80R mice.

The subsequent sections describe the option flags, file naming convention, and the invocation of the linker. "Link" and "Load" are used interchangably in this document.

SYNOPSIS

cc80 [option flag]... [file]...
(option flags and file names may be intermixed)

OPTION FLAGS

Being a UNIX shell script that invokes the Vandata compiler and the Nuvatec assembler/linker, cc80 tries to preserve the original option list as much as possible. However, a few modifications were unavoidable.

1) -s flag to cp1 is not optional.
2) -c, -i, -v flags to cp83 are obsolete.
3) +-maxcd flag to link defaults to +ca.
4) -e flag to cp83 has been changed to -j(for jump).
5) -i flag to link has been changed to -k(for keyboard).

Detailed description of each option flag can be obtained by running "man cc80", hence is omitted here.

FILE NAMING CONVENTIONS

Cc80 recognizes only 5 suffixes: c, l, s, o and v.

- .c for C source programs
- .v for Vandata assembly source (A-Sharp)
- .s for Nuvatec assembly source
- .o for object files
- .l for assembly listings

Additionally, the following files are generated by cc80.

a.out	linked-edited image
a.hex	a.out converted to Intel hex format
a.map	load map

Intermediate files whose suffixes are ".v", ".s" or ".l" will be automatically deleted upon completion of the command unless otherwise specified.

In case you want to save intermediate files, use
-kv for keep .v files
-ks for keep .s files
-kl for keep .l files

THE LINKER

As you will find, the linker from Nuvatec is quite versatile and as a matter of fact, it has its own command repertoire. Since these commands to the linker can't be easily contained in the cc80 argument list, you should either create a command file or type the commands at load-time. The corresponding flags are:

'-k'	reads link commands from the standard input.
'-f ⟨cmdfile⟩'	reads link commands from ⟨cmdfile⟩ .

Note that loading takes place if and only if either '-k' or '-f' is present in the argument list. Once 'a.out' is created , cc80 always makes 'a.hex', which is ready to be MICE'd.

EXAMPLES

```
% cc80 one.c
```
Compile and assemble 'one.c' but do not load

```
% cc80 one.c -fcmdfile
```
Compile and assemble 'one.c' and load according to the loader commands in the file ⟨cmdfile⟩. Also create 'a.hex'. Since the loader command file is supposed to be self-contained, this command does not necessarily imply that 'one.o' is to be loaded. Only those modules specified in ⟨cmdfile⟩ are loaded.

```
% cc80 -fcdmfile
```
Do not bother to compile anything. Just load. 'what' and 'how' ar specified in the file cmdfile .

```
% cc80 one.c -qv
```
Create 'one.v' by compiling 'one.c', then quit.

```
% cc80 one.c -bp
```
Compile and assemble 'one.c' using the brief protocol option.

```
% cc80 -k
```
The shortest command to invoke the linker. Loading will be done according to the loader commands you type in from your terminal.