

CROSSWARE[®] AZ80NT

Z80 Assembler for Windows

The Crossware AZ80NT is a sophisticated relocatable macro cross assembler that generates code for the Z80, Z80180 and HD64180 families of microprocessors. It comes complete with the Crossware Embedded Development Studio and runs under Windows 9x, Windows NT4.0, Windows 2000 and Windows XP.

HIGHLIGHTS

- High speed relocatable macro cross assembler
- Source level debug output in IEEE695 format
- Support for code and data bank switching
- Wide range of output file formats
- Highly user friendly Embedded Development Studio integrated development environment (see separate data sheet)

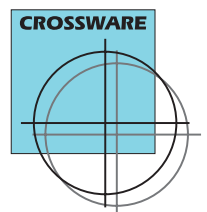
OVERVIEW

The package includes the following tools and utilities:

- relocatable macro cross assembler
- relocating linker
- make utility
- library manager
- Embedded Development Studio which includes
 - integrated project creation and maintenance utility
 - integrated editor
 - integrated terminal emulator
 - on-line books with full text search

(The Embedded Development Studio is described in detail in a separate data sheet)

All of these tools and utilities are Win32 applications built to make the most of your 32 bit computing environment. All tools can be used outside of the Embedded Development Studio if required using conventional command-line instructions.



Crossware Products
Old Post House
Silver Street
Litlington
Royston
Herts
SG8 0QE
United Kingdom

Telephone
+ 44 (0) 1763 853500

Facsimile
+ 44 (0) 1763 853330

Web
<http://www.crossware.com>

E-mail
sales@crossware.com

REF: AZ80NT/0302

TECHNICAL DETAILS

RELOCATABLE MACRO CROSS ASSEMBLER

The cross assembler converts assembler code into an intermediate object module format which is subsequently linked with other intermediate object modules to create the final program. Features include:

- High speed assembly to create relocatable object module
- Z80 and HD64180 mnemonics
- Nestable macros with full argument passing
- Nestable conditional assembly
- Comprehensive range of assembler directives and pseudo ops (see below)
- Complex expression evaluation
- Upper and lower case labels with up to 255 significant characters
- Upper and lower case opcode mnemonics
- Upper and lower case macro names with up to 255 significant characters
- Comprehensive error checking with descriptive error messages
- Debug output of symbolic information
- User definable listing format

RELOCATING LINKER

The linker combines object modules created with the compiler and/or the assembler to create the final code that will run on your target system. It carries out the following functions:

- scans each module to collect segment and variable information
- arranges and positions segments at appropriate memory locations to suit the memory organisation of the target system and any specific location information supplied by the user
- finalises the values of all variables and calculates the results of any incomplete expressions
- extracts and relocates the code from each module to produce the final target program

The target program can be produced in a number of different formats including Intel hex, Motorola S records, HP/Microtec IEEE695 format or as a binary rom image. (A utility is provided for users who wish to divide the binary rom image into separate files suitable for 8 bit wide memory chips).

The following assembler and psuedo-ops are available:

clist	equ	macr	nopage	sym
common	fail	mc	noseq	tll
defb	fields	md	nouna	una
defw	ifc	mex	obj	xdef
defm	ifnc	nam	opt	xref
defs	ific	noclist	org	Z80
defl	ifinc	nolist	page	64180
elsec	ifcc	nomc	pagewidth	
end	include	nomd	section	
endc	list	nomex	seqon	
endm	listf	noobj	spc	

An optional link map will show the final location and sizes of all segments and the values of all global variables.

DEBUG OUTPUT

The assembler can optionally generate full source level debug information. The linker updates these debug records to take account of the final location of the target program and outputs them to the target program file in IEEE695 format. Debuggers and in-circuit emulators that support this popular format can then use it to facilitate full source level debugging.

LIBRARY MANAGER

Instead of being used to create a final target program, the object modules produced by the compiler and assembler can be integrated into a library. The library manager performs the tasks of:

- combining object modules into a library
- adding modules to an existing library
- removing or extracting modules from an existing library
- listing the contents of a library

MAKE UTILITY

The MAKE utility simplifies the task of keeping object files, libraries and target programs up-to-date. It detects if any source or dependency files have changed since the last build and runs the appropriate tools (compiler, assembler, linker or library manager) to rebuild out-of-date files. It supports many advanced features including macros, inference rules, conditional inclusion and other preprocessing directives and in-line files with automatic temporary file creation.

Although the Embedded Development Studio uses its own integrated routines to keep projects up-to-date, this stand-alone MAKE utility can be used to build projects from the command-line or from within another application. The Embedded Development Studio will automatically create a makefile which is fully compatible with this stand-alone MAKE utility.

HOST SYSTEM REQUIREMENTS

IBM compatible PC with an Intel Pentium or above running Windows 9x, Windows NT4.0, Windows 2000 and Windows XP.