COSMIC, sarl

Version 3.32

October 1989

Cross Compiler User's Guide

for Z80/HD64180

XENIX-AT Host

COSMIC, sarl

Version 3.32

October 1989

Cross Compiler User's Guide

for Z80/HD64180

MS/PC-DOS Host

TABLE OF CONTENTS

PREFACE

CHAPTER 1

INTRODUCTION

CHAPTER 2

TUTORIAL INTRODUCTION

CHAPTER 3

PROGRAMMING FOR HD64180/Z80 ENVIRONMENTS

CHAPTER 4

USING THE COMPILER

CHAPTER 5

USING THE ASSEMBLER

CHAPTER 6

USING THE LINKER

CHAPTER 7

DEBUGGING SUPPORT

CHAPTER 8

USING THE PROGRAMMING SUPPORT UTILITIES

APPENDIX A

COMPILER ERROR MESSAGES

APPENDIX B

MODIFYING COMPILER OPERATION

APPENDIX C

HD64180/Z80 MACHINE LIBRARY

APPENDIX D

COMPILER PASSES

PREFACE

The Cross Compiler User's Guide for Z80/HD64180 is a reference guide for programmers writing C programs for Z80/HD64180 environments. It provides an overview of how the cross compiler works, and explains how to compile, assemble, link and debug programs on your host system for execution on your target system. It also describes the programming support utilities included with the cross compiler and provides tutorial and reference information to help you configure executable images to meet specific requirements. This manual assumes that you are familiar with your host operating system and with your specific target environment.

You can find information about Whitesmiths' implementation of the C programming language in your C Language Specification for Microcontroller Environments.

Organization of this Manual

This manual is divided into eight chapters and four appendixes.

Chapter 1, "Introduction," describes the basic organization of the C compiler and programming support utilities.

Chapter 2, "Tutorial Introduction," is a series of examples that show you step by step how to compile, assemble and link a simple C program using the C cross compiler.

Chapter 3, "Programming for Z80/HD64180 Environments," tells you how to use the features of C for your target processor to meet the requirements of your particular application. It explains how to create a runtime startup for your application, and how to write C routines that perform special tasks such as serial I/O, making direct references to hardware addresses, handling interrupts, and calling assembly language programs.

Chapter 4, "Using the C Compiler," describes compiler command line options and the action each option per-

forms. This chapter also describes the functions in the C runtime library.

Chapter 5, "Using the Assembler," describes the assembler and its use, as well as the command line options it accepts and what they do. It explains the rules that your assembly language source code must follow, and documents all directives that the assembler supports.

Chapter 6, "Using the Linker," describes the linker command line options and the action each option performs. This chapter also describes in detail all the features of the linker and how to use it to link your C and assembly language source files and object images.

Chapter 7, "Debugging Support," describes the support available for **cxdb,** the C source level cross debugger and for other debuggers or in-circuit emulators.

Chapter 8, "Using the Programming Support Utilities," describes each of the programming support utilities and the command line options each accepts. Examples of how to use these utilities both together and separately are also included.

Appendix A, "Compiler Error Messages," is a list of compile time error messages that the C compiler may generate.

Appendix B, "Modifying Compiler Operation," describes the "prototype file" that serves as input to the compiler command driver. It explains how to change the default operation of the compiler to suit your needs and how to write your own "programmable" command line options to the compiler driver. It also describes the compiler driver program, called **c.**

Appendix C, "Z80/HD64180 Machine Library," describes the assembly language routines that provide support for the C runtime library.

Appendix D, "Compiler Passes," describes the specifics of the preprocessor, parser, code generator and assembly language optimizer and the command line options that each accepts.

This manual also contains an **Index.**