

instant **Pascal**
user's manual

AIM 65
advanced interactive microcomputer



...where science gets down to business

© Rockwell International Corporation 1981
All Rights Reserved
Printed in the U.S.A.

Document No.
29650N85

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1	Introduction	
1.1	AIM 65 Instant Pascal Overview.....	1-2
1.2	AIM 65 Instant Pascal User's Manual Description.....	1-3
1.3	Reference Documents.....	1-5
2	Installation and Initialization	
2.1	Installing Instant Pascal ROMs.....	2-1
2.2	Entering, Exiting and Re-Entering Instant Pascal.....	2-5
	2.2.1 Entering Instant Pascal (5 Key)....	2-5
	2.2.2 Exiting Instant Pascal (ESC Key)...	2-5
	2.2.3 Re-entering Instant Pascal (5 Key).	2-6
	2.2.4 Start Application Program in PROM (6 Key).....	2-6
	2.2.5 Transitions Between Wait States....	2-6
3	Instant Pascal Operation	
3.1	Read Source Input (R Command).....	3-1
3.2	Source Output Commands.....	3-3
	3.2.1 List Current Text Unit (SPACE Command).....	3-3
	3.2.2 Move Text Pointer to the Top (T Command).....	3-3
	3.2.3 List Text Units (L Command).....	3-4
3.3	Initialization.....	3-6
	3.3.1 Report Available Memory (M Command).....	3-6
	3.3.2 Initialize Memory (N Command).....	3-7
	3.3.3 Initialize Instant Pascal (Z Command).....	3-7
	3.3.4 Redefine Page Width (W Command)...	3-7
3.4	Program Editing.....	3-8
	3.4.1 Move the Text Pointer to the Bottom (B Command).....	3-8
	3.4.2 Move the Text Pointer Up (U Command).....	3-9
	3.4.3 View Five Text Units (V Command)..	3-9
	3.4.4 Move the Text Pointer Down (D Command).....	3-9

TABLE OF CONTENTS (con't)

<u>Section</u>	<u>Title</u>	<u>Page</u>
	3.4.5 Delete Text Units (K Command).....	3-9
	3.4.6 Insert a Text Line (I Command)....	3-10
	3.4.7 Find a Text String (F Command)....	3-11
	3.4.8 Program Change Example.....	3-13
3.5	Program Binding (C Command).....	3-15
3.6	Program Execution.....	3-15
	3.6.1 Execute Program (G Command).....	3-15
	3.6.2 Execute and Trace Program (, Command).....	3-16
	3.6.3 Interrupt Program (DEL Command)...	3-17
4	Program Debugging	
4.1	Tracing.....	4-1
	4.1.1 Statement Tracing (S Command).....	4-1
	4.1.2 Assignment Tracing (A Command).....	4-2
4.2	Immediate Statement Execution (X Command).	4-2
5	Instant Pascal Text Units	
5.1	Text Units in Editing.....	5-2
5.2	Definitions of Text Units.....	5-7
6	Diagnostic Messages	
6.1	Source Input Diagnostics.....	6-1
6.2	Binder Diagnostics.....	6-3
	6.2.1 *SEQUENCE* Diagnostic.....	6-3
	6.2.2 Identifier Lookup Diagnostics.....	6-4
	6.2.3 Label Lookup Diagnostics.....	6-4
	6.2.4 Type Consistency Diagnostics.....	6-5
6.3	Execution-Time Diagnostics.....	6-6
6.4	Source Output Diagnostics.....	6-6
7	Instant Pascal Language Definition	
7.1	Summary of Extensions and Restrictions....	7-1
	7.1.1 Extensions.....	7-1
	7.1.2 Restrictions.....	7-2

TABLE OF CONTENTS (con't)

<u>Section</u>	<u>Title</u>	<u>Page</u>
7.2	Word-Symbols and Predefined Identifiers....	7-4
	7.2.1 Word Symbols.....	7-4
	7.2.2 Predefined Type Identifiers.....	7-4
	7.2.3 Predefined Boolean Constants.....	7-6
	7.2.4 Predeclared Procedures.....	7-6
	7.2.5 Predeclared Functions.....	7-10
7.3	Comparison to Standard Pascal.....	7-10
	7.3.1 Notation, Terminology and Vocabulary.....	7-10
	7.3.2 Identifiers, Numbers and Strings..	7-11
	7.3.3 Constant Definitions.....	7-11
	7.3.4 Data Type Definitions.....	7-11
	7.3.5 Declarations and Denotations of Variables.....	7-12
	7.3.6 Expressions.....	7-13
	7.3.7 Statements.....	7-13
	7.3.8 Procedure Declarations.....	7-14
	7.3.9 Predicates.....	7-14
	7.3.10 Input and Output.....	7-14
	7.3.11 Programs.....	7-15

TABLE OF CONTENTS (con't)

A	Summary of Commands.....	A-1
B	Language Summary Tables.....	B-1
C	Page 0 Memory Map.....	D-1
D	ASCII Character Set.....	E-1
E	Execution-Time Diagnostics.....	C-1
F	Selected Bibliography.....	F-1

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
2-1	AIM 65 Instant Pascal Memory Map.....	2-2

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
2-1	AIM 65 Instant Pascal ROM Addresses.....	2-4
2-2	State Transitions in Instant Pascal.....	2-7

Pascal is a powerful high level computer programming language originally designed for educational purposes. Developed by Niklaus Wirth of the ETH Technical Institute of Zurich in 1970 - 1971, Pascal has gained acceptance world-wide as a standard language to teach computer programming. The rich variety of Pascal language features allows a wide range of data structures to be specified and complex algorithms to be implemented. Programming in Pascal using structured programming techniques produces programs that are easy to write, understand and maintain. The wide-spread teaching of the language, coupled with the increased productivity of the programmer and the improved reliability of the generated program, is causing Pascal to be increasingly adopted in industrial and scientific applications as well as in the classroom.

The Pascal language defined by Niklaus Wirth is recognized as Standard Pascal and is referred to as such in this manual. Numerous world-wide installations of Pascal on main-frames, minicomputers, and microcomputers have resulted in many variations of the language in the form of extensions (added language features beyond Standard Pascal) and restrictions (Standard Pascal features that have not been implemented). These installations typically require a large amount of computer resources, e.g. up to 64K of RAM, a floppy disk, and a CRT-based terminal.

AIM 65 Instant Pascal* is a unique implementation of an extensive subset of Standard Pascal which combines the immediacy of ROM based system software, interactive debug facilities at the source code level, on-board AIM 65 printer and display peripherals along with low-cost expansion memory, to provide a complete Pascal educational, development and application system.

*Instant Pascal is a trademark of Melvin E. Conway

1.1 AIM 65 INSTANT PASCAL OVERVIEW

Instant Pascal is an implementation of the Pascal programming language for the Rockwell AIM 65 Microcomputer. The firmware is embodied in a five-ROM set which is installed by means of the AIM 65 expansion connector. Instant Pascal incorporates facilities to write and debug programs entirely at the source language level. These include source-level editor, statement and assignment trace and immediate source-statement execution for examination and modification of data.

Instant Pascal is a highly interactive programming tool which substantially extends the power of the AIM 65. A major subset of the Standard Pascal language, it incorporates all of the simple and structured statements, and the most widely used simple and structured data types. Extensions to the language permit direct control of memory-mapped I/O of the microcomputer systems and allow interfacing to machine-language programs developed with the AIM 65 assembler.

Features of Instant Pascal include:

- All editing occurs at the source language level
- Translation at program input and output obviates any need to know internal program formats
- Lister contains a built-in prettyprinter which displays program and data structure
- Source programs may be conveniently modified during the program testing process
- Source-level trace lists statements as they are executed

- Independent assignment trace lists values as they are changed
- Immediate statement execution permits operator examination and modification of data
- String extensions to language facilitate development of interactive programs

AIM 65 Instant Pascal implements a substantial subset of the Standard Pascal defined in the Jensen and Wirth book "Pascal User Manual and Report" (see Appendix F). Extensions to their standard included in AIM 65 Instant Pascal and language features in the standard but not included in AIM 65 Instant Pascal are described in detail in Section 7.

AIM 65 Instant Pascal is both a compiler and an interpreter. It compiles source statements written in Pascal directly from the keyboard, from the AIM 65 Text Editor or from external media (audio tape or user defined) into an internal format. The internal format contains both the object code to be interpreted upon execution and source code identifiers for formatted printout to support debugging and editing at the source code level.

1.2 AIM 65 INSTANT PASCAL USER'S MANUAL CONTENTS

Section 1, Introduction, introduces the AIM 65 Instant Pascal product with language features, extensions and restrictions summarized. Common Pascal language and programming books are also referenced.

Section 2, Installation and Initialization, tells how to install the AIM 65 Instant Pascal ROMs and how to enter, re-enter and exit Instant Pascal.

Section 3, Instant Pascal Operation, explains how to operate AIM 65 Instant Pascal in conjunction with the AIM 65 Debug Monitor and Text Editor. This includes program input, output, editing, checking and execution.

Section 4, Program Debugging, describes how to use Instant Pascal commands to debug a program written in Pascal. This includes statement and assignment tracing, as well as immediate statement execution.

Section 5, Instant Pascal Text Units, explains how the Pascal Source code is treated for the purpose of editing, checking and execution.

Section 6, Diagnostic Messages, describes the meanings of diagnostic messages that may occur during translation, execution and listing as well as the effect on Instant Pascal operation.

Section 7, Instant Pascal Language Definition, defines the language extensions and restrictions with respect to Standard Pascal.

Appendix A, Summary of commands, summarizes all commands used for source program entry, editing and saving, as well as debugging and execution.

Appendix B, Language Summary Tables, includes several summary tables of language words, symbols and capabilities.

Appendix C, Page 0 Memory Map, defines variables that are alterable by the user.

Appendix D, ASCII Character Set, lists the hexadecimal and decimal codes for letter, number and special characters as well as control commands.

Appendix E, Execution-Time Diagnostics, defines the meanings of error codes displayed during Pascal program execution.

Appendix F, Selected Bibliography, lists several reference books on Pascal and programming.

1.3 REFERENCE DOCUMENTS

Users of this manual should have the manuals supplied with their AIM 65, particularly the AIM 65 User's Guide (29650 N36), the R6500 Programming Manual (29650 N30) and the R6500 Hardware Manual (29650 N31).

Readers who are unfamiliar with Pascal are advised to read one of the many tutorials on this language (See Appendix F). These three books are especially recommended:

- o Cherry, George W., Pascal Programming Structures. Reston Publishing Company, Reston, VA, 1980.
- o Grogono, Peter, Programming in Pascal. Addison-Wesley Publishing Company, Reading, MA, 1980.
- o Fox, D. and Waite, M., Pascal Primer. Howard W. Sams & Co. Inc., Indianapolis, IN, 1981.

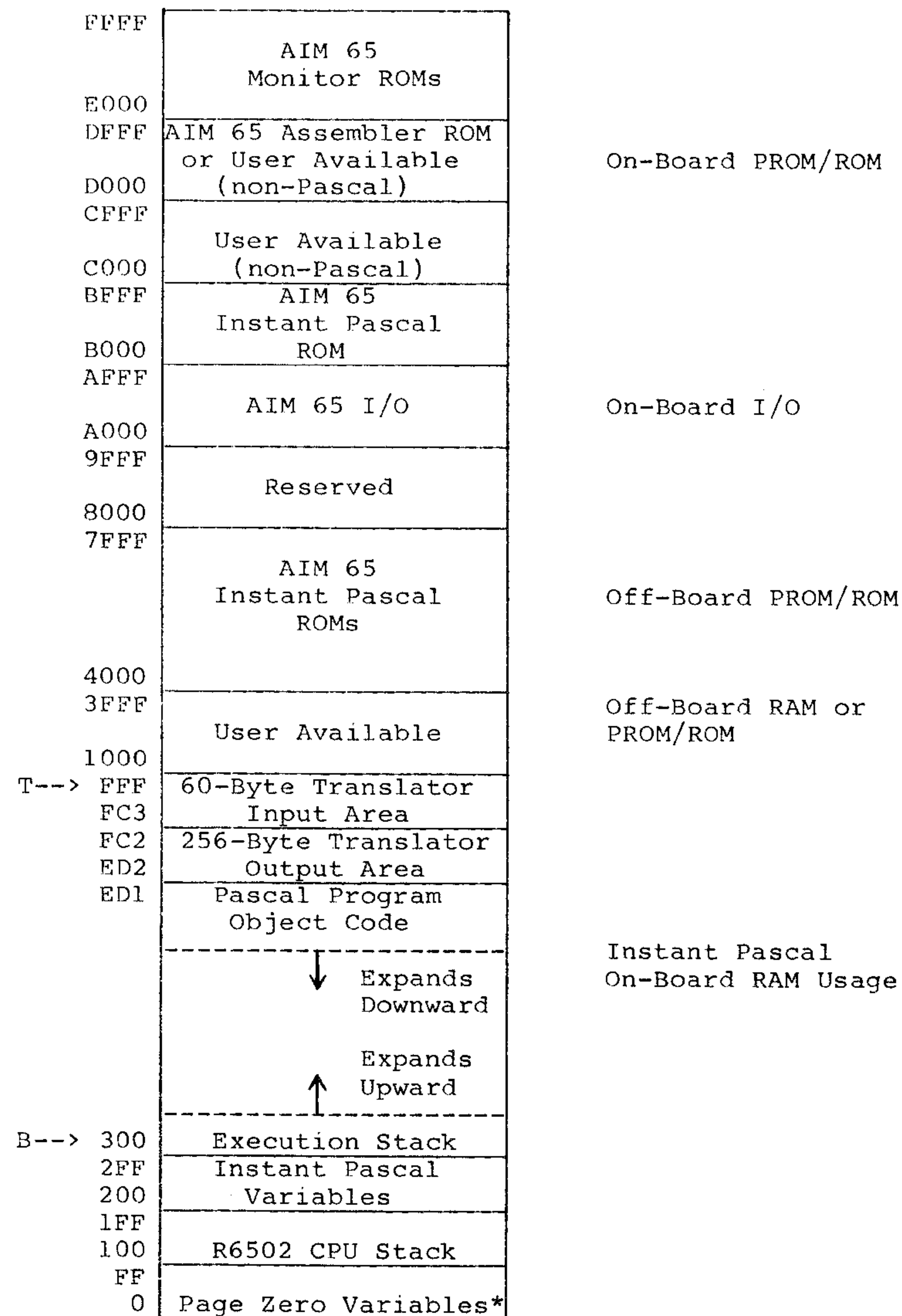
SECTION 2

INSTALLATION AND INITIALIZATION

The AIM 65 Instant Pascal is provided in five Rockwell R2332 4K-byte ROMs. Four of the ROMs must be installed in a PROM/ROM module external to the AIM 65 while the other one may be installed on-board the AIM 65 Master Module. Figure 2-1 shows the memory map of AIM 65 Instant Pascal. Note that Instant Pascal operates in conjunction with the AIM 65 Monitor, therefore the Monitor ROMs must be installed. Since Instant Pascal can link to machine code, the optional AIM 65 Assembler may reside on-board concurrently with Instant Pascal, but is not required. After installing the Instant Pascal ROMs and connecting the PROM/ROM module to AIM 65, Instant Pascal is ready for operation.

2.1 INSTALLING OF INSTANT PASCAL ROMS

Before removing the ROMs from the shipping package, be sure to observe the handling precautions listed in Section 1.4 of the AIM 65 User's Guide. Since MOS devices may be damaged by the inadvertent application of a high voltage, be sure to discharge any static electrical charge accumulated on your body by touching a ground connection (e.g. a grounded equipment chassis) before touching the ROMs or the AIM 65. This precaution is especially important if you are working in a carpeted area or in an environment with low relative humidity. Ensure that power is turned off to the AIM 65. Carefully remove any PROM or ROM device that is installed in socket Z26 on the AIM 65 Master Module. Remove the ROMs from the shipping package and verify that the pins are straight and free from foreign



T = Top of the user program including Translator input and output areas (default value = \$0FFF)

B = Bottom of the user-program (default value = \$0300)

* See Appendix D

Figure 2-1. AIM 65 Instant Pascal Memory Map

material. Install the ROMs on the AIM 65 Master module and on the PROM/ROM Module in accordance with ROM address ranges listed in Table 2-1. You can install R32P6 on-board the AIM 65 in socket Z26, however the other ROMs must be installed off-board. After the ROMs are installed in the PROM/ROM module, connect the module to the AIM 65 Expansion connector in accordance with the PROM/ROM or adapter module user instructions.

The 4K bytes of AIM 65 on-board RAM provide 2764 bytes available for the application program after subtracting memory dedicated to page 0, page 1 and AIM 65 Instant Pascal overhead (see 3.3.1). This allows a Pascal program of about 1800 source characters in length to be entered when memory is dedicated to the Pascal program object code, i.e. compiler input is from the keyboard or external media, or about 1100 source characters when the compiler input is from memory, i.e. the text buffer in the AIM 65 Text Editor. For larger programs, add off-board expansion RAM from \$1000 - \$3FFF.

NOTE

It is recommended that a Rockwell RM 65 16K PROM/ROM Module be used installed in an RM 65 card cage and connected to the AIM 65 through an RM 65 Adapter/Buffer Module. Additional RAM can easily be added using RM 65 8K Static RAM modules or an RM 65 32K Dynamic RAM module. The RM 65 modules that may be used and their part numbers are:

- | | |
|-----------|-------------------------------|
| RM65-3216 | 16K PROM/ROM Module* |
| RM65-3132 | 32K Dynamic RAM Module |
| RM65-7004 | 4-Slot Piggyback Module Stack |
| RM65-7008 | 8-Slot Card Cage |
| RM65-7016 | 16-Slot Card Cage |
| RM65-7101 | Single Card Adapter* |
| RM65-7104 | Adapter/Buffer |
| RM65-7116 | Cable Driver Adapter/Buffer |

*Minimum Expansion Set for AIM 65 Instant Pascal operation.

Table 2.1. AIM 65 Instant Pascal ROM Addresses

ROM No.	Address	Module (1)	Socket
R32P2	4000-4FFF	RM 65 PROM/ROM	Z12 (2)
R32P3	5000-5FFF	RM 65 PROM/ROM	Z13 (2)
R32P4	6000-6FFF	RM 65 PROM/ROM	Z14 (2)
R32P5	7000-7FFF	RM 65 PROM/ROM	Z15 (2)
R32P6	B000-BFFF	AIM 65 Master Module	Z26

NOTES

(1) ROMs may be installed in any PROM/ROM module that operates with R2332 ROMs in the corresponding address range.

(2) Recommended sockets - ROMs may be installed in any socket that can be configured for the corresponding address ranges.

2.2 ENTERING, EXITING, AND RE-ENTERING INSTANT PASCAL

2.2.1 Entering Instant Pascal (5 Key)

Press 5 to enter Instant Pascal from the AIM 65 Monitor when the Monitor prompt "<" is displayed. When power has just been turned on, or when Instant Pascal has not been previously entered, the AIM 65 will respond with the first of two initialization questions:

<5> MEMORY SIZE?

Press RETURN to enter the default value of 4096. If more contiguous RAM than 4K bytes is available and you wish to take advantage of it, enter the number of bytes in decimal, up to a maximum of 16384. The second question,

WIDTH?

requests the page width for program listings and data output. Press RETURN to enter the default value of 20 for the AIM 65 printer. If the KB/TTY switch is, or is about to be, set to TTY (see AIM 65 User's Guide, 1.9.2), first enter the number of printer columns in decimal. (This value may be changed later by using the +<W> command in Instant Pascal; see 3.3.4).

2.2.2 Exiting Instant Pascal (ESC Key)

Any time Instant Pascal is in an input wait state, pressing ESC will return control to the AIM 65 Monitor. This can occur when the AIM 65 is waiting for an Instant Pascal command after the "+<" prompt or when an Instant Pascal program is waiting for input data. Also, a listing produced by the +<L> command or a program trace may be aborted and control returned to the AIM 65 Monitor by holding down the ESC key until the printing stops.

2.2.3 Re-entering Instant Pascal (5 Key)

Whenever Instant Pascal is exited to the AIM 65 Monitor, you can return to Instant Pascal without loss of the program in memory, provided that the RAM occupied by the program is not disturbed. Just press 5 when the Monitor prompt is displayed. Instant Pascal checks locations \$60 and \$61 for a value which is set up during the initialization sequence. If this value is present, the initialization sequence is bypassed and the Instant Pascal prompt "+<" is displayed. When this happens, Instant Pascal has assumed that the integrity of RAM storage has not been disturbed. If RAM occupied by the Instant Pascal program was altered while the AIM 65 Monitor was in control, the results are unpredictable. This can be avoided (at the cost of the Pascal program in memory) by reinitializing memory after entering by means of the +<N> command (see 3.3.2).

2.2.4 Start Application Program in PROM (6 Key)

When the AIM 65 Monitor is waiting for a command, pressing 5 causes the Monitor to execute a JMP to \$B000, which enters Instant Pascal. Pressing 6 executes a JMP to \$B003. The Instant Pascal ROM contains a JMP \$C000 in \$B003, so that Key 6 may be used to enter a function at \$C000-\$D000 from the Monitor. A Pascal program is limited to \$300-\$3FFF, however (see Figure 2-1).

2.2.5 Transitions Between Wait States

Table 2-2 expands upon Section 2.2.2 by describing certain transitions which may be forced by other keys as well as ESC.

Table 2-2. State Transitions in Instant Pascal

Cmd	State	Key	Effect
I	Waiting for source input	ESC RETURN	Returns to AIM 65 Monitor Completes input, command prompt
R	Waiting for source input	ESC RETURN (2)	Returns to AIM 65 Monitor Completes input, command prompt
G	Executing	DEL(held)	Breaks to command prompt
	Waiting for data input	ESC	Returns to AIM 65 Monitor
	Tracing	ESC(held) SPACE(held)	Returns to AIM 65 monitor Pause; any key resumes
L	Listing	ESC(held) SPACE(held)	Returns to AIM 65 Monitor Pause; any key resumes