ELLIOTT 900

Volume   2:   PROGRAMMING INFORMATION

Part   5:   SIR SYSTEMS

Section 1:   INTRODUCTION TO 900 SIR SYSTEMS ROUTINES

CONTENTS

Chapter 1: GENERAL INFORMATION

1.1 Introduction

The SIR Systems routines are written to provide standard programs and a consistent environment for various commonly used operations within SIR programs. They are used by various 900 series standard software programs, and should be used wherever possible when writing any SIR program.

The user programming in SIR may choose not to use any of the standard software, other than the Assembler itself. However he does in most cases use various subroutines from the program library described in Volume 2. Volume 2, Part 5 describes routines which can provide a standard framework or environment for the user in this case. The user of Algol and Fortran has these facilities automatically available, as they are 'built-in' to the compilers and their operating systems.

The types of operation covered by these routines include: Input and output of characters, Interrupt routines, Error Outputs and Termination of programs; also Floating Point and Double Length operations.

Wherever practical, new library software programs and major re-issues of existing programs will be compatible with the SIR systems philosophy.

1.2 Purpose

The reason for introducing the SIR systems routines is to avoid proliferation of programs with individual 'built-in' input-output, error routines, etc. By defining a set of standard sub-routines the programs that use them may be made more flexible. For example a new peripheral or non-standard code may be used for output, by altering the character output sub-routine only.

The individual routines are described in the remainder of Part 5.

1.3 Reserved areas of Store

Locations 8 to 31 of the store will be reserved for Systems use. Locations 8 to 20 should not be used by the programmer, except in ways specifically defined by the descriptions of Software programs in Volume 2. Issues of SIR following Issue 2 will normally commence assembling programs at 32, and the programmer should avoid explicit reference to locations 21 to 31.

These locations are reserved to simplify communication between programs, particularly where extra store or peripherals are required.

Future peripherals for 900 series machines may also use fixed store locations, near the beginning of the first store module.

Chapter 2:   USE OF STANDARD ROUTINES

   2.1   Standard Packages

           Volume 2 Part 5 describes two types of standard routines:

   (a)   Special packages for floating point operations, double
           length fixed point, etc.

   (b)   Standard routines for character input, errors,
           interrupts, etc.

           Type (b) are in fact specifications of interfaces.   The
programmer should write his program to the interface described.   At least
one standard routine is supplied to match the "other side" of the interface.
For different conditions, stores or peripherals, different routines may be
supplied or written by the user to match the same interface.

           The user is recommended to make up a package of the systems
routines he commonly used.   This package may be made up into a relocatable
binary tape, which will save a considerable amount of computer time in
assembling a complete set of programs, particularly where the package
includes QF or QDLA.

   2.2   Relocatable Binary Packages

           To make up the relocatable binary tape, input option zero (*0)
by entering SIR at 8.   (Use the on-line teleprinter or a small piece of paper
tape).   All the other mnemonic tapes required for the package are then input
by entry at 9 to SIR, followed by a 'newline % newline' (input by entry at 9,
similarly to the option).

           When assembling individual programs, assemble by entering
SIR at 8, with any continuation tapes at 9.   Input a 'newline % newline halt code'
sequence on the teleprinter or from a piece of paper tape, by entering at 9.
EU messages for the missing labels will be output.   Load the relocatable
binary package and enter at 11.   The final 'FIRST LAST NEXT' message
should not be preceded by any EU messages.