IDENTIFICATION

Product Code:	DEC-08-BA1C-D
Product Name:	PDP-8, 8S, 8I, 8L System Program Index
Date Created:	October 16, 1968
Maintainer:	Program Library

DEC-08-AJAC-D FOCAL

FOCAL (for FOrmula CALculator) is an on-line, conversational, service program for the PDP-8 family of computers, designed to help scientists, engineers, and students solve numerical problems. The language consists of short imperative English statements which are relatively easy to learn. Mathematical expressions are typed, for the most part, in standard notation. No previous programming experience is needed either to understand this manual or to use FOCAL at the Teletype console. However, the best way to learn the FOCAL language is to sit at the Teletype and try the commands, starting with the examples given in the manual.

DEC-08-AFAC FORTRAN Compiler and Operating System

The one pass FORTRAN Compiler and Operating System compiles FORTRAN source language statements into an object program tape. The operating system executes the program. This operating system contains the interpreter, arithmetic function subroutines and input/output packages.

DEC-08-AFA2 Symbol Print

Loaded over the FORTRAN Compiler, this program lists the variables used and where they will be located in core. It also indicates the section of core not used by the compiled program and data.

DEC-08-A2A0 FORTRAN (8K)

Describes a more powerful version of FORTRAN designed specifically for the PDP-8 family of computers with 8K words of core memory. The manual details the FORTRAN language elements and statements, compiler and operating system, Symbolprint, Linking Loader, Run-Time Monitor, and the complete library of arithmetic subprograms. The appendices include a summary of library calling sequences, format and statement specifications, storage allocation, error diagnostics, and operating procedures for the entire system.

DEC-08-ASAB PAL III Symbolic Assembler

PAL III is a two pass symbolic machine language assembler which converts program coded in symbolic machine language to binary machine language. It has an optional third pass to produce a side-by-side octal/symbolic assembly listing. The basic process performed by the assembler is the substitution of numeric values for symbols, according to associations defined in the symbol table. In addition, the user may request that the assembler itself assign values to the user's own symbols at assembly time. These symbols are normally used to name memory locations, which may then be referenced by name.

DEC-08-CDDA DDT-8 (Dynamic Debugging Tape)

DDT-8 provides a means for on-line program debugging at the symbolic or mnemonic level. By typing commands on the console teleprinter, memory locations can be examined and changed, program tapes can be inserted, selected program tapes can be inserted, selected portions of the program can be run, and the updated program can be punched.

DEC-08-CMAA MACRO-8 Assembler

The MACRO-8 symbolic assembler accepts source programs written in symbolic language and translates them into binary form in two passes. MACRO-8 produces an object program tape (binary), a SYMBOL table (for use with DDT), an Octal/Symbolic assembly listing, and useful diagnostic messages. MACRO-8 is compatible with PAL III, and has the following additional features: user defined macros; double precision integers, floating point constants; arithmetic and Boolean operators, literals, text facilities and automatic link generation.

DEC-08-COC0 ODT-8 (Octal Debugging Technique)

ODT-8 (Octal Debugging Technique) is a debugging aid for the PDP-8, which facilitates communication with, and alteration of, the program being run. Communication between operator and program occurs via the Teletype, using defined commands and octal numbers. ODT-8 is a subset of DDT-8 and occupies three pages of core storage.

The program may be relocated to occupy any three consecutive pages of core.

DEC-08-ESAB Symbolic Editor

The PDP-8 Symbolic Editor allows the user to prepare and edit symbolic tapes on-line in ASCII code with the Teletype and/or high-speed reader/punch. The tedious task of correcting symbolic program tapes using the Teletype off-line is thereby avoided. Proper use of the PDP-8 Symbolic Editor can substantially ease the labor and reduce the number of passes necessary to correct symbolic program tapes.

The Editor reads a page, or section, of symbolic tape into a buffer in core storage, where it is available for examination and correction upon keyboard command. The page buffer occupies all of core not taken up by the Editor itself and has a capacity of approximately 6000_{10} characters. When the Editor has finished reading a page into the buffer, a bell rings to signal the user that he may begin editing. The user may then call for a listing of individual (numbered) lines, in any order, and insert desired changes and corrections. In addition, text may be added to the buffer, or inserted between specified lines. Groups of lines or individual lines may be moved or deleted by a single command, or the entire page may be erased if desired. Searches may be made and parts of lines changed without retyping the entire line. Upon keyboard command, the Editor will then either list or punch out the corrected lines or page on paper tape. The Editor can also be used to generate a new symbolic tape by typing new text directly on the keyboard. Errors in typing may be corrected simply by typing a rubout.

DEC-LB-SYYB Lab 8 Averager

Conversational package to do signal averaging; back averaging, dual resolution averaging. Computation of point to point standard deviation and trend; signal sorting features; 1024 data points; preset sweep counting. User may do on line signal editing.

DEC-08-NGCB Console Manual

DEC-08-UDCA Calculator-8

The calculator is a program written to evaluate FORTRAN-like equations. It differs from FORTRAN in that functions to be evaluated are entered via the keyboard and calculated immediately upon termination of the entry. Format control and the ability to call common function subroutines are provided.

F-85 PDP-8 Users Handbook

DEC-08-YQYA Floating-Point System

Includes Floating-Point Interpreter and I/O subsystems. Allows the programmer to code his problem in floating-point machine language.

Floating-point operations automatically align the binary points of operands, retaining the maximum precision available by discarding leading 0s. In addition to increasing accuracy, floating-point operations relieve the programmer of scaling problems common in fixed-point operations. This system includes elementary function subroutines programmed in floating-point. These subroutines are sine, cosine, square root, logarithm, arc tangent, and exponential functions. Data being processed in floating-point is maintained in three words of memory (12-bit exponent, 24-bit mantissa). An accuracy of seven decimal places is maintained.

Digital-8-16-S Master Tape Duplicator

This program will duplicate and verify 8-channel paper tapes using the PDP-8 with high-speed reader and high-speed punch. The program uses the program interrupt and allows both the reader and the punch to operate at maximum speed.

The program accumulates two types of checksums while reading and while punching: (1) the number of nonzero characters on the tape, and (2) the sum of characters on the tape (both are taken modulo 4096).

While duplicating, the program compares the checksums at the end of the tape with the checksums accumulated by the read routine. If these differ, a reader error has occurred and a message is typed.

Tapes are verified by reading them and comparing accumulated checksum with those at the end of the tape. Only master tapes produced by the program may be duplicated. The master tape has the two checksums punched at the end.

FUNCTION SUBROUTINES

DEC-08-FMAA Single Precision Square Root

This subroutine will extract the square root of a single precision integer. Given an input N ($0 < N < 2^{12}$), it will produce an integer K and a remainder R, such that $N = K^2 + R$.

DEC-08-FMBA Single Precision Signed Multiply Subroutine

This subroutine forms a 22-bit signed product from 11-bit signed multiplier and multiplicand.

DEC-08-FMCA-D Single Precision Signed Divide Subroutine

The Single-Precision Divide Subroutine will divide a 12-bit signed divisor into a 24-bit signed dividend to produce a 12-bit quotient and a 12-bit signed remainder.

DEC-08-FMDA Signed Double Precision Multiply

This subroutine forms a 46-bit signed product from the 23-bit signed multiplier and multiplicand.

DEC-08-FMEA Double Precision Signed Divide Subroutine

The Double-Precision Divide Subroutine will divide a 24-bit signed divisor into a 48-bit dividend to produce a 24-bit signed quotient and an unsinged remainder.

DEC-08-FMFA Double Precision Sine Subroutine

The Double-Precision Sine Subroutine will evaluate the function Sin(X) for $-4 < X < 4$ (X is in radians). The argument is a double-precision word, 2 bits representing the integer part and 21 bits representing the fractional part. The result is a 23-bit signed fraction $-1 < \text{Sin}(X) < 1$.

DEC-08-FMGB Double Precision Cosine Routine

This subroutine will form the cosine of a double-precision argument (in radians). The input range is $-4 < X < 4$.

DEC-08-FMHA Four Word Floating Point Package

This program is almost identical to the 3-Word Floating-Point Package (Digital-8-5-S) except that accuracy is carried to 35 bits, and four 12-bit words are used for storage.

DEC-08-FMIA Logical Subroutines

Subroutines for performing the logical operations of inclusive and exclusive OR are presented as a package.

DEC-08-FMJA Arithmetic Shift Subroutines

Four basic subroutines, shift right and shift left each at both single and double precision, are presented as a package. These are arithmetic shifts.

DEC-08-FMKA Logical Shift Subroutines

Two basic subroutines, shift right at both single and double precision are presented as a package. These shifts are logical in nature.

Digital-8-21-F Signed Single-Precision Multiply Subroutine for Extended Arithmetic Element Type 182

This subroutine forms a 22-bit signed product from an 11-bit signed multiplier and multiplicand using the Extended Arithmetic Element Type 182. It occupies less storage and takes less time to execute than its non-EAE counterpart and it has the same calling sequence.

Digital-8-22-F Signed, Single-Precision Divide Subroutine Using the Extended Arithmetic Element Type 182

This subroutine will divide a double-precision signed 22-bit dividend by a signed 11-bit divisor producing a signed 11-bit quotient and an 11-bit remainder having the sign of the dividend.
It makes use of the Extended Arithmetic Element Type 182 instruction set and occupies less storage and takes less time to execute than its non-EAE counterpart. It has the same calling sequence except that the subroutine name has been changed from DIVIDE to SPDIV.

Digital-8-23-F Signed Double Precision Multiply Subroutine Using the Extended Arithmetic Element Type 182

This subroutine will multiply a 23-bit, signed 2s complement binary number by a 23-bit signed 2s complement binary number giving a 46-bit product with two signs on the high order end. It makes use of the Extended Arithmetic Element Type 182 instruction set and therefore occupies fewer core locations and takes less time to execute than its non-EAC counterpart. Its calling sequence is compatible with the non-EAE version.

Digital-8-25-F Floating Point Package Using the Extended Arithmetic Element Type 182

These packages perform the same tasks as the Floating-Point Packages (Digital-8-5-S A, B, C, D) except that certain routines have been speeded up by the use of the Extended Arithmetic Element Type 182.

LOADERS

DEC-08-LRAA Read-In Mode (RIM Loader)

The RIM Loader is a minimum routine for reading and storing information contained in read-in-mode coded tapes via the ASR33 Perforated Tape Reader.

DEC-08-LBAA Binary Loader

The Binary Loader is a short routine for reading and storing information contained in binary-coded tapes, using the ASR33 Perforated-Tape Reader and the Type 750 High-Speed Perforated Tape Reader.

The Binary Loader accepts tapes prepared by the use of PAL (Program Assembly Language) or MACRO-8. Diagnostic messages may be included on tapes produced when using either PAL or MACRO. The Binary Loader will ignore all diagnostic messages.

DEC-08-LHAA "HELP" Loader

The "HELP" Loader loads the standard version of the RIM and BIN Loaders into the PDP-8, in less than 90 s, replacing manual procedures which required several minutes.

DEC-08-LUAA TC01 Bootstrap Loader

This is a bootstrap for loading the PDP-8 DECtape Library System designed for use with DECtape Control Type TC01 with TU55 Tape Transports.

Digital-8-3-U DECtape Library System Loader (Type 552 Control)

The use of the DECtape Library System Loader is discussed. Certain conventions with respect to last page storage are established for this loader as well as the Read-In-Mode and Binary Loaders.

UTILITY PROGRAMS AND SUBROUTINES

DEC-08-PMPO Read-In Mode (RIM) Punch

The RIM Punch program provides a means of punching out information contained in selected blocks of core memory as RIM-coded tape via the ASR33 Perforated Tape Punch or High-Speed Punch. The punch program may occupy either low or high memory depending on the version used.

Digital-8-5-U-Sym Binary Punch (ASR33 or PC01)

This program provides a means of punching out information contained in selected blocks of core memory as binary-coded tape via the ASR33 Perforated Tape Punch or via the high-speed punch.

Digital-8-6-U-Sym Octal Memory Dump

This routine will read the console switches twice to obtain the upper and lower limits of an area of memory, then type on the Teletype an absolute address plus the octal contents of the first four words specified and repeat this until the block is exhausted, at which time the user may repeat the operation.

Digital-8-10-U-Sym BCD to Binary Conversion Subroutine

A basic subroutine for converting binary-coded-decimal numbers to their equivalent binary value. Conversion is accomplished by "radix deflation."

Digital-8-11-U Double Precision BCD to Binary Conversion Subroutine

This subroutine converts a 6-digit BCD number to its equivalent binary value in two computer words.

Digital-8-14-U-Sym Binary to BCD Conversion Subroutine

This subroutine provides the basic means of converting binary data to binary-coded-decimal (BCD) data for typeout, magnetic tape recording, etc.

Digital-8-15-U-Sym Binary to BCD Conversion (Four Digit)

This subroutine extends the method used in Digital-8-14-U-Sym so that binary integers from 0 to 4095 contained in a single computer word may be converted to four binary-coded-decimal characters packed in two computer words.

Digital-8-17-U-Sym Extended Arithmetic Element Type 182 Instruction Set Simulator

NO tape

This simulator will allow the use of instructions belonging to the Extended Arithmetic Element Type 182 set on a PDP-8 that is not equipped with this hardware option. Programs that will ultimately be run on a PDP-8 with the EAE option can be checked out using the simulator on a machine without EAE. All that is required to use the EAE hardware when it becomes available is a reassembly of the program without the simulator symbolic tape.

Digital-8-18-U-Sym Alphanumeric Message Typeout

A basic subroutine to type messages packed in computer words. Two 6-bit characters are packed internally in a single word. All ASR33 codes from 301 to 337 and from 240 to 277 (excepting 243 and 245) can be typed. The typing of line-feed (code 212) and carriage-return (code 215) are made possible by arbitrarily assigning internal codes of 43 and 45, respectively, to represent these characters, thus preventing the output of ASCII codes 243 (#1) and 245 (%).

Digital-8-19-U-Sym Teletype Output Subroutines

A group of subroutines useful in controlling ASR33 output is presented as a package. Provision is made for the simulation of tabulation stops. The distance "tabbed" may be controled by the user. Characters whose ASR33 codes are in groups 241 through 277, inclusive, and 300 through 337, inclusive, are legal. Space, carriage return then line feed, and tabulation are provided via subroutines.

Digital-8-20-U-Sym Character String Typeout

A basic subroutine to type messages stored internally as a string of coded characters. All ASR33 characters are legal.

Digital-8-21-U-Sym Symbolic Tape Format Generator

The format generator allows the user to create PDP-8 symbolic tapes with formatting. It may be used to condense tapes with spaces by inserting tabs, or merely to align tabs, instructions, and comments.

Digital-8-22-U-Sym Unsigned Decimal Print

This subroutine permits the typeout of the contents of a computer word as a 4-digit, positive, decimal integer.

Digital-8-23-U-Sym Signed Decimal Print - Single Precision

This subroutine permits the typeout of the contents of a computer word as a signed 2s complement number. If bit 0 of the computer word is a 1, the remaining bits represent a negative integer in 2s complement form; if bit 0 equals 0, the remaining bits represent a positive integer. If the number is negative, a minus sign is printed; if positive, a space.

Digital-8-24-U-Sym Unsigned Decimal Print, Double Precision

This subroutine permits the typeout of a double-precision integer stored in the usual convention for double-precision numbers. The one exception is that all 24 bits are interpreted as magnitude bits (i.e., the bit 0 of the high-order word is not a sign bit). The typeout is in the form of a 7-digit, positive, decimal integer.

Digital-8-25-U-Sym Signed Decimal Print, Double Precision

This subroutine permits the typeout of the contents of two consecutive computer words as one signed, double-precision, 2s complement number. If bit 0 of the high-order word is a 1, the remaining 23 bits represent a negative integer in 2s complement form; if bit 0 equals 0, the remaining bits represent a positive integer. If the number is negative, a minus sign is printed; if positive, a space.

Digital-8-28-U-Sym Single Precision Decimal to Binary Conversion and Typeout ASR33, Signed or Unsigned

This routine accepts a string of up to four decimal digits (single precision for the PDP-8) from the Teletype keyboard and converts it to the corresponding 2s complement binary number.

- M List

The string may contain as legal characters a sign (+,-, or space) and the digits from 0 through 9. If the first legal character is not a sign, the conversion is unsigned. A back arrow ( ) at any point in the string erases the current string and allows the operator to reenter the correct value. Any character after the first, other than another digit or back arrow causes the conversion to terminate and is found in location SISAVE within the subroutine.

Digital-8-29-U-Sym Double Precision Decimal to Binary Conversion and Typeout (ASR33), Signed or Unsigned

This routine accepts a string of up to eight decimal digits (double-precision for the PDP-8) from the Teletype keyboard and converts it to the corresponding 2s complement binary number.

The string may contain as legal characters a sign (+, -, or space) and the digits 0 through 9. If the first legal character is not a sign, the conversion is unsigned. A "back-arrow" ( ) at any point in the string erases the current string and allows the operator to reenter the value. Any character after the first, other than another digit or "back-arrow", causes the conversion to terminate and is found in location DIDSAV within the subroutine.

DECTAPE SYSTEM SOFTWARE

DEC-08-SUBO DECtape Programming Manual

PDP-8 DECtape software provides the programmer with the following operational materials for both the 552/555 hardware and the TC01/TU55 hardware.

a. Subroutines which the programmer may easily incorporate into a program for data storage, logging, data acquisition, data buffering (queueing), etc.

b. A library calling system for storing named programs on DECtape and a means of calling them with a minimal size loader.

Digital-8-33-U 5/8 Tog (DECtape Formatter) for 552/555 DECtape Control

The main purpose of this program is to supply a blank DECtape with a skeletal structure of required timing-track marks and mark-track codes for a specified number of blocks, each with a specified number of data words. A series of end-zone marks are also supplied at each end of the tape. Creating a tape in the above manner requires three full length passes, involving one normal halt. Following these three passes is a series of four check-read passes and one check-write pass involving only one normal halt at the end of the last pass. Data is written both in the forward and reverse directions and each write pass is followed by two check-read passes, one in each direction.

For a complete certification of the DECtape format produced, the use of the exerciser program DECEX (Digital-8-33B-U) is recommended.

Digital-8-34-U PDP-5/8 DECtape Exerciser (DECEX) (552 Control)

DECEX is a program in four discrete operative sections. This program exercises a DECtape unit in read data, write data, and search modes. DECEX operates equally well on either a PDP-5 or PDP-8, requiring no special adjustment by the user. The four discrete program sections are EXER, SEXER, ALTER, and INK.

EXER exercises the DECtape in read data, write data, and search modes. Exercising is accomplished by writing a known data record on tape in segments of 22_{10} blocks and then reading this record back into memory and checking it against the original, word for word. Comparison errors as well as errors detected automatically by the control are reported to the user via written output to the ASR33. Control over the operational configuration is available to the user via the SWITCH REGISTER at all times.

SEXER exercises the search mode only. The user specifies a unit number and block number to the program which then searches for the specified block alternating between forward and reverse. The user may change the target block number at any time during the operation of the program. Errors are reported by an error halt.

ALTER and INK are utility programs which allow the user to vary the data pattern written on tape by EXER.

Only tapes constructed in the standard format may be exercised by EXER. Any format is compatible with SEXER.

DEC-08-FUA0 TC01 DECtape Subroutines

These subroutines provide the user with the ability to read, write and search using the TC01 tape system. The read and write subroutines transfer 128_{10} (one memory page) of the specified block (or blocks) although the standard block length is 129_{10} 12-bit words. Successive blocks are read (written) from (into) successive 128 word blocks of core. Provision is made for transfers to and from extended memories.

DEC-08-EUFA TC01/TU55 DECtape Formatter

The purpose of this system is to record the required timing and mark tracks on a DECtape mounted on the TC01-TU55 DECtape unit.

The program, which never stops, obtains the variable information it needs by communication with the operator via the ASR33 Teletype.

Two full passes are required to complete one DECtape. Upon completion of a sequence, another tape may be mounted and formatted, as the last, without renewed communication between the operator and program. Therefore, marked tape may be produced in great numbers with little operator intervention, at a rather rapid rate. One tape, excluding tape setup time, requires two minutes from start to finish (see also Disc Software).

DATA COMMUNICATION SYSTEM (Type 680)

Digital-8-35-S-A 680 5-Bit Character Assembly Subroutines

These subroutines concentrate Teletype data by assembling serial bit data into 5-bit characters and presenting the user with data similar to that obtained by using a 630 and scanner. They also add start and stop bits to 5-bit characters and transmit them in serial-bit fashion. Full-duplex lines are assumed, but the subroutines can operate with half duplex if the user handles the expected echo.

Digital-8-35-S-B 680 8-Bit Character Assembly Subroutines

The 680 Data Communication System 8-Bit Character Assembly Subroutines concentrate Teletype data by assembling serial-bit data into 8-bit characters and present the user with data similar to that obtained by using a 630 DCS and scanner. They also add start and stop bits to 8-bit characters and transmit them in serial-bit fashion. Full-duplex lines are assumed, but the subroutines will work with half duplex if the user handles the expected echo.

Digital-08-USBO Multianalyzer Display and Analysis

The two-dimensional pulse-height analysis program reads in and analyzes two-parameter energy and spectra data. The program receives and executes commands from the keyboard. These commands start and stop data taking, control the displays, and control writing and punching of data. The displays available are: isometric, vertical and horizontal slicing, differential and integral contours, and "twinkle box." The program is flexible with respect to the dimensions of the data matrix.

DEC-08-G61D Programmed Buffered Display 338 Manual

The Type 338 Programmed Buffered Display is a precision incremental display system, consisting of a small scale, high-speed computer and a display subsystem for control of the CRT. The computer used is the Digital Equipment PDP-8 (for Programmed Data Processor). It is a single address, fixed word length (twelve bits) machine. The complete cycle time for its random access magnetic core memory is 1.5 μs. All arithmetic operations are performed in 2s complement notation.

Digital-8-12-S Incremental Plotter Subroutine

This subroutine moves the pen of an incremental plotter to a new position along the best straight line. The pen may be raised or lowered during the motion.

Digital-8-15-S Oceanographic Analysis

This program represents the basic accepted physical oceanography method for the reduction of data concerning depth, temperature, and salinity measurements of the water column.

This program allows the field oceanographer a rapid means of immediately calculating Sigma-T, anomaly of specific volume, and sound velocity following a Nansen cast whereby he may examine results in detail to determine the structure of the environment he has just sampled and to check the validity of his measurements.

The program also contains an interpolation routine as well as a depth integration of the anomaly of specific volume.

DISC/DECTAPE SOFTWARE

DEC-D8-SDAA DISC MONITOR SYSTEM

This system consists of a keyboard-oriented Monitor, which enables the user to efficiently control the flow of programs through his PDP-8, and a comprehensive software package, which includes a FORTRAN Compiler, Program Assembly Language (PAL-D), Edit program (EDITOR), Peripheral Interchange Program (PIP) and Dynamic Debugging Technique (DDT-D) program. Also provided is a program (BUILDER) for generating a customized monitor according to the user's particular machine configuration (amount of core, number of discs or DECtapes, etc.).

The system is modular and open ended, permitting the user to construct the software required in his environment, and allows the user full access to his disc (or DECtape)-referred to as the system device - for storage and retrieval of his programs. By typing appropriate commands to the Monitor, the user can load a program (construct it from one or more units of binary coding previously punched out on paper tape or written on the disc by the Assembler, and assign it core), save it (write it out, with an assigned starting address, on the system device), and later call it (read it back into core from the system device) for execution.

In order to have a complete DISC/DECtape package, the user may order the following in addition to DEC-D8-SDAA-D above:

1.	Disc System Builder	DEC-D8-SBAC-PB
2.	Disc Editor	DEC-D8-ESAB-PB
3.	PIP	DEC-D8-PDAP-PB
4.	Disc DDT	DEC-D8-CDDO-PB

5.	Disc DDT Driver (ASCII)	DEC-D8-CDDO-PA
6.	Disc/DECtape FORTRAN	DEC-D8-AFA(1-6)-PB
7.	PAL-D Assembler	DEC-D8-ASAA-D (Manual) DEC-D8-ASAA-PB

DEC-D8-ASAA PAL-D DISC ASSEMBLER

PAL-D is the symbolic assembly program designed primarily for the 4K PDP-8 family of computers with disc or DECtape. The PAL-D Assembler makes machine language programming easier, faster, and more efficient. Basically, the Assembler processes the programmer's source program statements by translating mnemonic operation codes to the binary codes needed in machine instructions, relating symbols to numeric values, assigning absolute core addresses for program instructions and data, and preparing an output listing of the program, which includes notification of any errors detected during the assembly process.