Software Product Description

PRODUCT NAME: **VAX BLISS-32 Implementation Language, Version 4.6** **SPD 25.12.18**

DESCRIPTION

BLISS-32 is a high-level, systems implementation language for VAX systems. BLISS-32 supports development of modular software according to structured programming concepts by providing an advanced set of language features. BLISS-32 provides access to most of the hardware features of the VAX systems to facilitate programming of real-time/hardware dependent applications.

BLISS-32 is intended for the development of operating systems, compilers, run-time system components, data base file systems, communications software, utilities, and so forth, for use on a VAX hardware system.

The BLISS-32 compiler runs in native mode under the VMS Operating System. It translates BLISS-32 source programs into relocatable object modules which can be linked for execution. BLISS-32 compiled code is optimized for execution efficiency.

The following features of BLISS-32, sometimes described in VAX or VMS terminology, are machine-independent concepts. Collectively, this set of common features is known as ''Common BLISS'' and can be used in a transportable manner to develop programs for more than one computer architecture.

Common BLISS Features

- Modules for modularity and convenient development are separately compiled. Object modules are relocatable and can be linked with object modules produced by the VAX MACRO assembler or other native mode languages.
- BLISS-32 provides expressions for describing the actions to be performed and declarations for allocating, describing and initializing data, defining macros and literals, and so forth.
- ''Type-free'' - All data is manipulated as longwords of 32 bits. Interpretation of data is provided by language operators.
- Operators provide a set of operations for integer and floating point arithmetic, comparison, maximization and minimization of signed integer, unsigned integer and address values, and Boolean operations.
- Field references - Allow values to be retrieved from or assigned to any contiguous field of 1 to 32 bits located anywhere in the VAX virtual address space.
- Character sequence functions - Provide efficient run-time manipulation of character data. Operations include moving, concatenating, comparing, and translating character sequences, as well as searching for particular characters or substrings of characters.
- IF, CASE, SELECT, and SELECTONE - Allow the choice of an expression or group of expressions to be executed based on programmed tests.
- DO, WHILE, and UNTIL - Allow general loops that cycle as long as a programmed test is satisfied. The test can be made either at the beginning or the end of the loop.
- INCR and DECR - Allow counted loops that execute a computed number of times under control of a loop variable.
- LEAVE - Allows both early termination of the processing of a named block and continuation after the named block. LEAVE can be considered a restricted form of forward-only GOTO since there is no general GOTO in BLISS-32.
- OWN and GLOBAL - Provide static storage allocation; GLOBAL names are made available to the Linker and resolve EXTERNAL data declarations in other modules.
- LOCAL, STACKLOCAL, and REGISTER - Allow dynamic stack-like allocation using either the execution stack or the general registers.
- INITIAL and PRESET provide for initialization of static and dynamic storage.
- STRUCTURE - Allows programmed definition of arbitrary data structures in terms of an accessing algorithm for locating elements of a structure. Built-in declarations for VECTORs, BLOCKs, BITVECTORs, and BLOCKVECTORs provide for commonly-needed structures.
- ROUTINE - Provides subroutines or procedures in BLISS-32. Routines are recursive, re-entrant and can be linked in sharable images for use by multiple processes.

- REQUIRE - Allows source text from other files to be automatically included in the module being compiled.
- LIBRARY - Allows special compiler-produced binary declaration files to be included in the module being compiled. The effect is substantially the same as REQUIRE however, it is more efficient because a restricted set of declarations can be precompiled into internal form.
- MACRO and KEYWORDMACRO - Allow compile-time definition of both positional and keyword oriented macros. Macro definition and replacement are in terms of source lexical units lexemes (atoms, tokens) rather than character text. Macro calls and macro declarations can be nested and recursive.
- %IF, %THEN, %ELSE, and %FI - Allow conditional compilation of BLISS source depending on programmed, compile-time tests. These can be used to control expansion of macros or in their own right.
- Lexical Functions - Allow a variety of compile-time operations such as concatenation of strings, construction of names, testing properties of macro parameters, testing compiler qualifiers, generating compiler diagnostic messages, and controlling macro expansion.

The following features of BLISS-32 are specialized for use on VAX systems. These features provide a precise means to tailor BLISS-32 programs to the special capabilities of the VAX and VMS environment.

- LINKAGE - Allows definition of specialized calling sequences for time critical or unusual applications. Options allow for: use of CALLS/CALLG/RET or JSB/BSB/RSB type call and return instructions, for passing parameters in VAX general registers or in parameter blocks, controlling the preservation and use of registers by a routine, and the sharing of registers across a set of routines as high-speed, common storage. Built-in linkage declarations for BLISS and FORTRAN fully support the VAX calling sequence conventions.
- PSECT - Allows use of link-time program sections for efficient use of the virtual address space. By default, generated code sections are position independent.
- BUILTIN - Allows use of VAX machine-specific functions for access to VAX features not otherwise provided by the BLISS-32 language. Machine-specific functions generally correspond to VAX-11 instructions such as: ADAWI, BISPSW, CRC, HALT, INDEX, MTPR, PROBER, REMQUE, and so forth. Over 80 such functions are provided.
- ENABLE, together with SIGNAL, SIGNAL__STOP, and SETUNWIND functions - allow use of the VMS condition handling and error message reporting mechanisms.

As a native-mode language, BLISS-32 is integrated into the VAX Common Language Environment. This integration provides VAX BLISS-32 users with:

- Support for VAX interlanguage calling standard
- Access to all VMS system services
- Access to the facilities of the VAX Symbolic Debugger
- VAX Language-Sensitive Editor support
- VAX Source Code Analyzer support
- Callable interfaces to the VAX Common Run-time Library
- Callable interfaces to VAX utilities such as SORT and optional products (for example, VAX DATATRIEVE)

The BLISS-32 compiler performs global and local optimization to produce efficient and compact generated code. Each routine definition is treated as a complete unit in compiling the code for that routine. While industry terminology varies, the following generally characterizes the optimizations employed: common subexpression elimination, removal of loop invariants, constant folding, block register allocation, peephole replacement, test instruction elimination, jump verses branch instruction resolution, branch chaining, and cross-jumping.

The compiler will optionally check for certain language usage that, while technically correct, may be a coding error.

The BLISS-32 compiler optionally produces a listing file that shows the source text compiled and the generated code in a format that closely resembles a VAX MACRO assembly listing. Multiple listing options allow selective inclusion or exclusion of source generated code, use of source names in the listing in place of machine register names (where feasible), macro expansion and tracing information, and use of names from library files.

A detailed symbol cross-reference is optionally included in the listing.

HARDWARE REQUIREMENTS

VAX, MicroVAX or VAXstation configuration as specified in the System Support Addendum (SSA 25.12.18-x).

BLISS-32 compiler performance varies with the virtual memory working set for which it is allowed. In typical:

- With large-scale compilation of complex programs, a working set allocation of over 400 pages (200K bytes) can be desirable for efficient system utilization.
- In an environment where only a single BLISS compilation occurs at any time, a system configuration of two megabytes of memory is required.
- In a multi-user environment, where several concurrent compilations occur, a minimum of two megabytes is recommended per simultaneous compilation.

SOFTWARE REQUIREMENTS*

For systems using terminals: (No DECwindows interface)

- VMS Operating System

For workstations running VWS:

- VMS Operating System
- VMS Workstation Software

For workstations running DECwindows:

VMS Operating System (and necessary components of VMS DECwindows)

* Refer to the System Support Addendum (SSA 25.12.18-x) for availability and required versions of prerequisite/optional software and for information regarding components of VMS DECwindows.

ORDERING INFORMATION

Software Licenses: QL-106A*-**
Software Media: QA-106A*-**
Software Documentation: QA-106AA-GZ
Software Product Services: QT-106A*-**

* Denotes variant fields. For additional information on available licenses, services and media, refer to the appropriate price book.

SOFTWARE LICENSING

This software is furnished under the licensing provisions of Digital Equipment Corporation's Standard Terms and Conditions.

For more information about Digital's licensing terms and policies, contact your local Digital office.

LICENSE MANAGEMENT FACILITY SUPPORT

This layered product supports the VMS License Management Facility. License units for this product are allocated on a CPU-capacity basis.

For more information on the License Management Facility, refer to the VMS Operating System Software Product Description (SPD 25.01.xx) or the License Management Facility manual of the VMS Operating System documentation set.

For more information about Digital's licensing terms and policies, contact your local Digital office.

SOFTWARE PRODUCT SERVICES

A variety of service options are available. For more information on these or other services, please contact your local Digital office.

SOFTWARE WARRANTY

Warranty for this software product is provided by Digital with the purchase of a license for the product as defined in the Software Warranty Addendum of this SPD.