VAX

1981

ARCHITECTURE HANDBOOK

digital

DIGITAL Facility, Boylston, Massachusetts

CORPORATE PROFILE

Digital Equipment Corporation designs, manufactures, sells and services computers and associated peripheral equipment, and related software and supplies. The Company's products are used world-wide in a wide variety of applications and programs, including scientific research, computation, communications, education, data analysis, industrial control, timesharing, commercial data processing, word processing, health care, instrumentation, engineering and simulation.

ARCHITECTURE HANDBOOK

digital

Digital Equipment Corporation makes no representation that the interconnection of its products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting of license to make, use, or sell equipment constructed in accordance with this description.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

This handbook was designed, produced, and typeset
by DIGITAL's New Products Marketing
using an in-house text-processing system.

CONTENTS

Designing the VAX Architecture

Several years ago, foreseeing the explosive growth of minicomputer uses, DIGITAL committed numerous corporate resources and personnel to the goal of creating a computer family for the 1980s and beyond. Extending and, at the same time, protecting our customers' enormous investment in PDP-11 computers and software lay at the heart of our efforts. But simultaneously we wanted to extend virtual address space to a degree that would eliminate the need for overlays and segmentation of programs. We also wanted to increase the capability and ease of use of our new computers, so that more and more applications could be handled by people with less and less specialized training. Along with this, we aimed for a computer that would serve a wider range of applications (OEM, laboratory realtime, distributed data processing, interactive, and so on) than any minicomputer from any other vendor.

What resulted from these efforts was the VAX family of 32-bit virtual memory minicomputers: high-speed, easy-to-use, highly dependable machines destined to satisfy our customers' needs for a decade and more. Central to these computers is the VAX architecture, an architecture that meets four critical design goals beyond that of maximal compatibility with the PDP-11s. We wanted high bit efficiency, and achieved it through an extensive collection of data types and addressing modes; we wanted a systematic, elegant instruction set, and achieved one with independence of operators, data types, and addressing modes, one that is easily exploited, particularly by high-level languages.

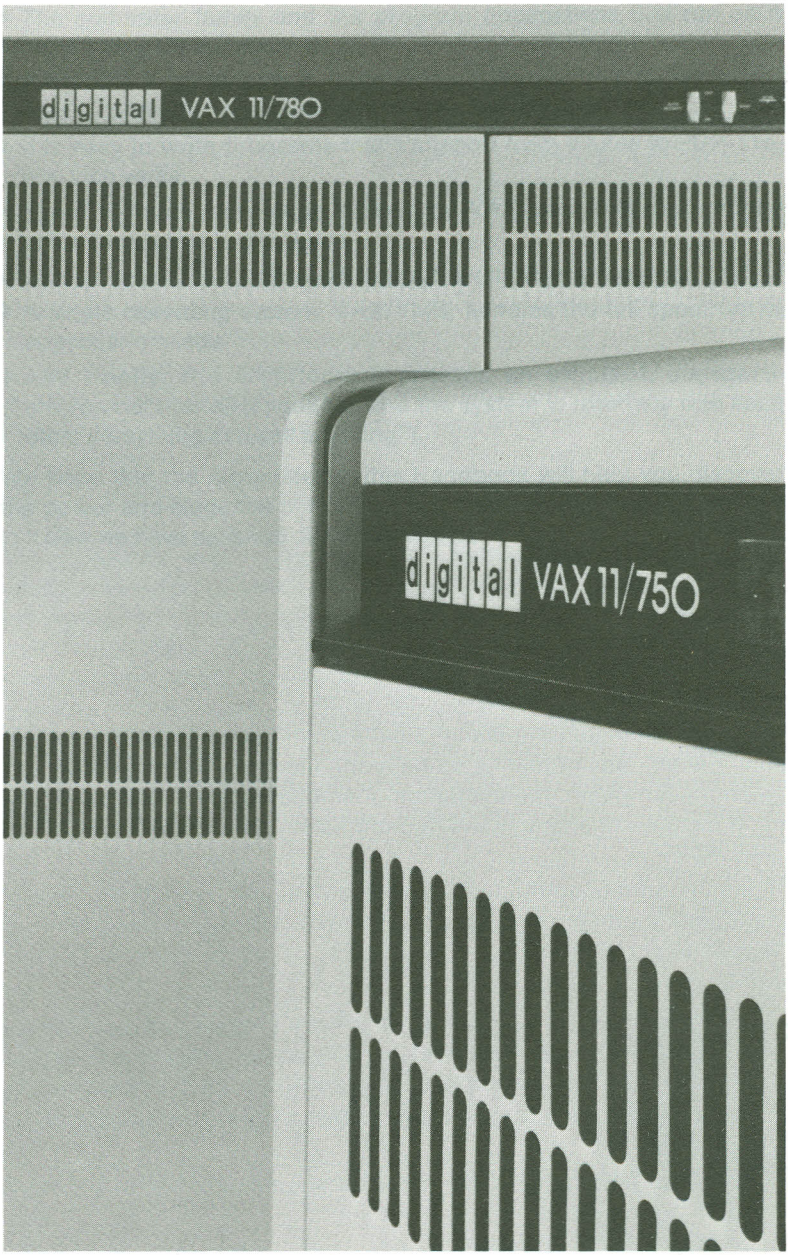
We wanted extensibility: new operators and data types may be added consistently with existing ones. And finally, we wanted a single architecture that would span and be suitable for all members of the VAX family, present and future.

In fulfilling these goals, DIGITAL engineers designed an architecture that provides you with numerous benefits above and beyond the high interactivity and friendliness our computers have always stood for:

- Programming time is minimized, so that programmer productivity is increased.
- Hardware peripherals are compatible with existing PDP-11 computers.

- Realtime response is significantly improved.
- The computer family and the program applications that run on it have a long life expectancy.
- Programs are completely transportable across VAX family members — providing adequate options and file spaces.
- It is easy to write programs that transport from VAXes to PDP-11s, and vice versa.
- There is much more code sharing on VAX machines than on any other computers in existence.
- System reliability is improved through architectural features.
- A single operating system, VAX/VMS, handles the full spectrum of application needs.
- And finally, the architecture creates an efficient, standard environment for all languages and the system to interface with each other ("anything can call anything").

We hope that the remainder of this Handbook will help you discover the power and elegance of the VAX architecture, and that you will see in it how we have satisfied our design goals.


digital VAX 11/780
digital VAX 11/750

CHAPTER 1

VAX: COMPUTERS FOR THE '80s

The next decade will witness ever-widening, perhaps unpredictable, demands upon computers and the computer industry. In finance, government, industry, and possibly even in the home, computers will serve expanding roles, solving problems, managing processes, or facilitating communication. DIGITAL has developed an innovative computer technology to confront the challenges of the 1980s, a technology that offers vast power and enormous flexibility for every kind of application. At the same time, we have held fast to the philosophy of affordability and easy use that made DIGITAL the minicomputer industry leader.

VAX is the name of our innovative computer family. In the short time since the introduction of the first family member, VAX has proven itself as one of the friendliest, and at the same time, most powerful, computer designs available. Its continual enhancement with new processors, peripherals, high-level languages, and other sophisticated software, positions VAX as the computer family able to meet the needs of the future.

Virtual Address Space

The letters *VAX* suggest the premier feature of VAX computers—Virtual Address eXtension. In a VAX computer, bytes of information are located with a 32-bit address. This means effectively that the computer can recognize more than four billion addresses, a vast amount in minicomputer and programmer's terms. The remarkable thing about this giant "address space" is the it is *virtual*: the (physical) main memory of the computer need not be anywhere near as large as four billion bytes for the machine actually to process data whose addresses are scattered through the address space. In fact, what happens is that a sophisticated scheme called "memory management" allows programmers to operate as if a big part of the virtual address space were really available to them, and then it handles all the details of storing programs and subsequently bringing them into main memory where they are processed.

From the programmer's point of view, the bottom two billion bytes of virtual address space can be used for programs, and he or she need never worry about complicated techniques of overlaying or segmenting to squeeze the program into a smaller address range. Logic built right into the VAX computers quickly translates all the programmer's virtual addresses into physical addresses, stores the

programs and data in convenient locations (disks or main memory) and brings into main memory whatever parts of the program or data are needed at any instant.

Another aspect of memory management is the rapid switching of "contexts." VAX is a high-powered multiprocessor: many programs and many programmers can use it simultaneously, each appearing to own unique control of the processor. Actually, the computer is processing the programs—or pieces of them—one at a time, and switching into and out of main memory the "context" (loosely speaking, the environment) of many programs. A switched-in context allows a program to run; a switched-out context makes the program wait for the central processor. Consequently, numerous different activities could be occurring on a VAX computer at any one time: a data acquisition procedure, a long computation project, an editing session, an interprocess communication; and the context switching takes place so swiftly that each user would feel like the only user.

Scientific, industrial, commercial, and educational market users have already put the original VAX model through its paces in numerous situations: realtime, computational, program development. In the upcoming decade we will see a wide range of new usages handled by VAX computers.

At the heart of the VAX computer family is its architecture. For our purposes, architecture is the collection of attributes common to all family members, attributes that guarantee that all software runs without change on all family members. Particularly pertinent are the instruction set, the memory management algorithms, and certain other aspects of the design that help define contexts and processes.

A distinction should be made between the architecture and the implementation of that architecture. For example, the architecture of the typewriter is essentially fixed: it is the keyboard layout; knowing the alphabet and punctuation systems, any typist can make it work, can "process" jobs. Each manufacturer may, however, implement that architecture in individual ways. Some may have striking print keys, some may have typing head balls; some may have a blue keyboard, some black. In addition, the builder could trade off one feature against another: a lighter touch vs. the capability to make numerous carbons. Nevertheless, all machines still serve the essential function, typing.

Similarly with computer architecture. Each processor in the family may bear slightly differing implementations and tradeoffs; yet all will fulfill the core of requirements put on the machine by the designers, and all will deliver the same service to the users. Once having learned the instruction set, for example, a programmer is ensured that exactly

the same instruction will perform precisely the same operation on each processor in the VAX family.

VAX architecture is appropriate over a variety of system costs, performance and application needs. Therefore, a huge range of user requirements can be met, at a lower cost, since the price of supporting many different architectures is eliminated.

Two very important aspects of the VAX architecture are its power and its connection to another major DIGITAL computer family, the PDP-11.

The most obvious manifestation of the VAX architecture is the instruction set. Over three hundred instructions give the assembly level programmer extensive control of computer operation. Each instruction has a mnemonic, a shorthand name that suggests its job. (Obvious ones are ADD, DIV, MOV, and PUSH). *Orthogonality* (i.e. independence) is incorporated into the instruction set. That is, the operation being performed (e.g., ADD), the type of data used (e.g., longword), and the method of addressing (e.g., autodecrement) can all be considered independently by the compiler. This makes for faster, more efficient, and easier to implement compilers.

In addition, each instruction operates on its "natural" number of operands, from zero up to as many as is appropriate. Also, some recurrent operations from high-level languages are engineered into the hardware, so that a single instruction can handle them. The FORTRAN DO loop and three-operand addition (A = B + C) are examples of operations that can be handled by a single VAX instruction. Finally, there is no forced alignment on longword boundaries: as required by many languages, data items bigger than a byte can still reside on any byte boundary.

The architecture also includes instructions to make various applications and operating system codes more efficient. There are, in this group, hardware support of queues, easy access to variable length bit fields, and simple instructions to save or restore a program context.

Because DIGITAL foresaw the possibility of more and more applications, the architecture is extendible. The instruction set can be expanded efficiently to include new data types and operators in a way that consistently matches all the ones that already exist. Enormous flexibility is assured this way, since what exists now does not significantly constrain what may be added in the future.

We use the word *compatibility* to designate VAX's connection to the PDP-11 family of minicomputers from DIGITAL. Customers have a large investment in the PDP-11 computers and software. To protect that investment, and to simplify the procedures by which program-

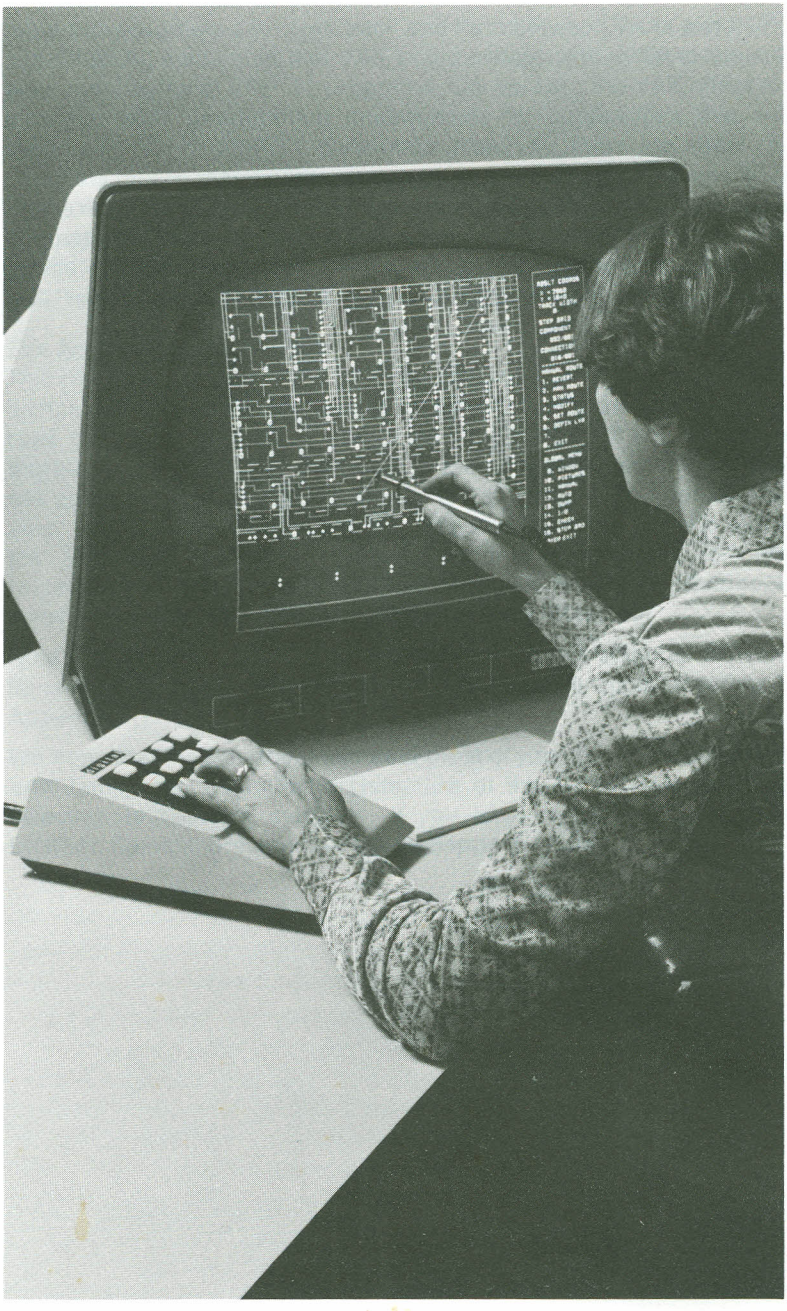
mers and programs can move back and forth between VAXs and PDP-11s, DIGITAL made sure that VAX would accept, with minimal conversion, most types of PDP-11 programs. Conversely, the VAX offers an excellent host development environment for applications that will eventually run on PDP-11 computers. Naturally, there are some restrictions, but in many cases, simple recompilation of programs is all that is required to carry a PDP-11 program to the VAX. VAX even has a compatibility mode at the hardware level, so that many PDP-11 programs can run unchanged on it. Compatibility mode may run along with "native" mode programs in a VAX multiprocessing environment.

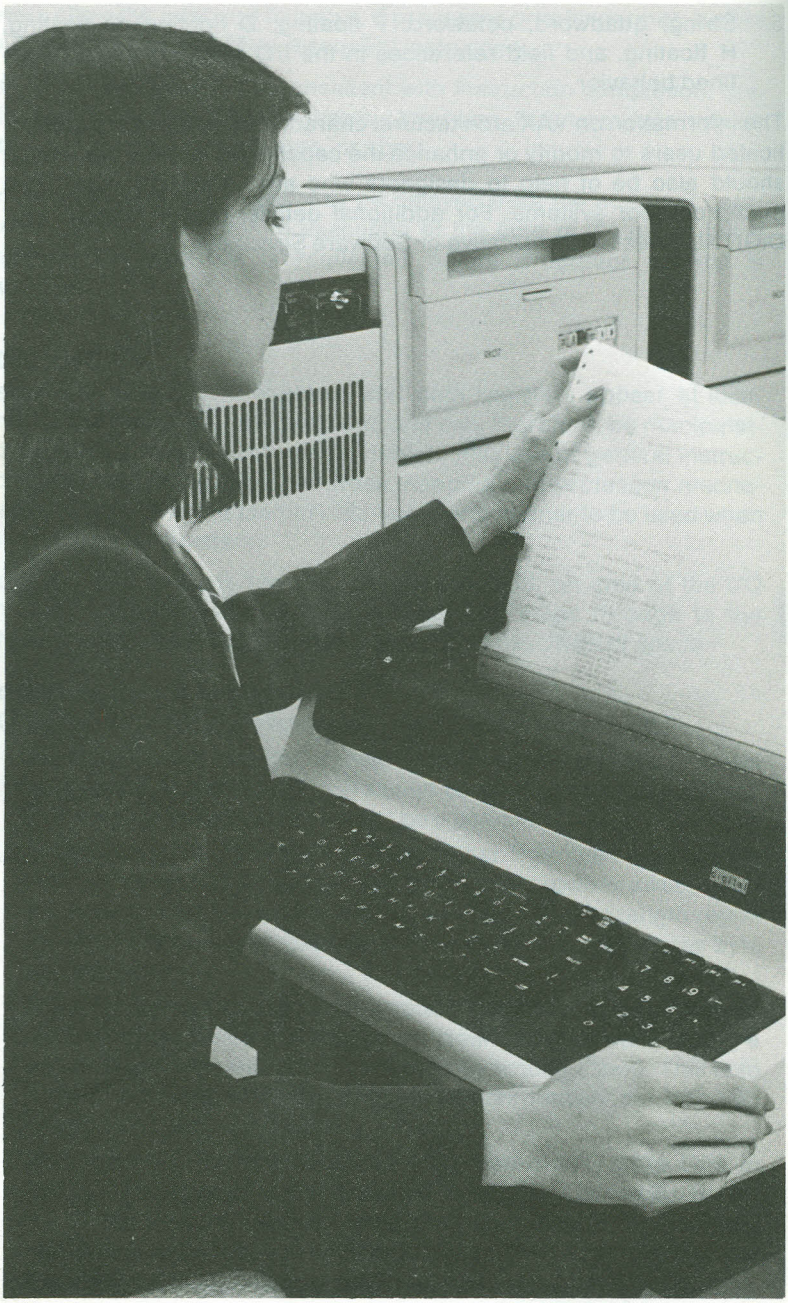
The Architecture Handbook

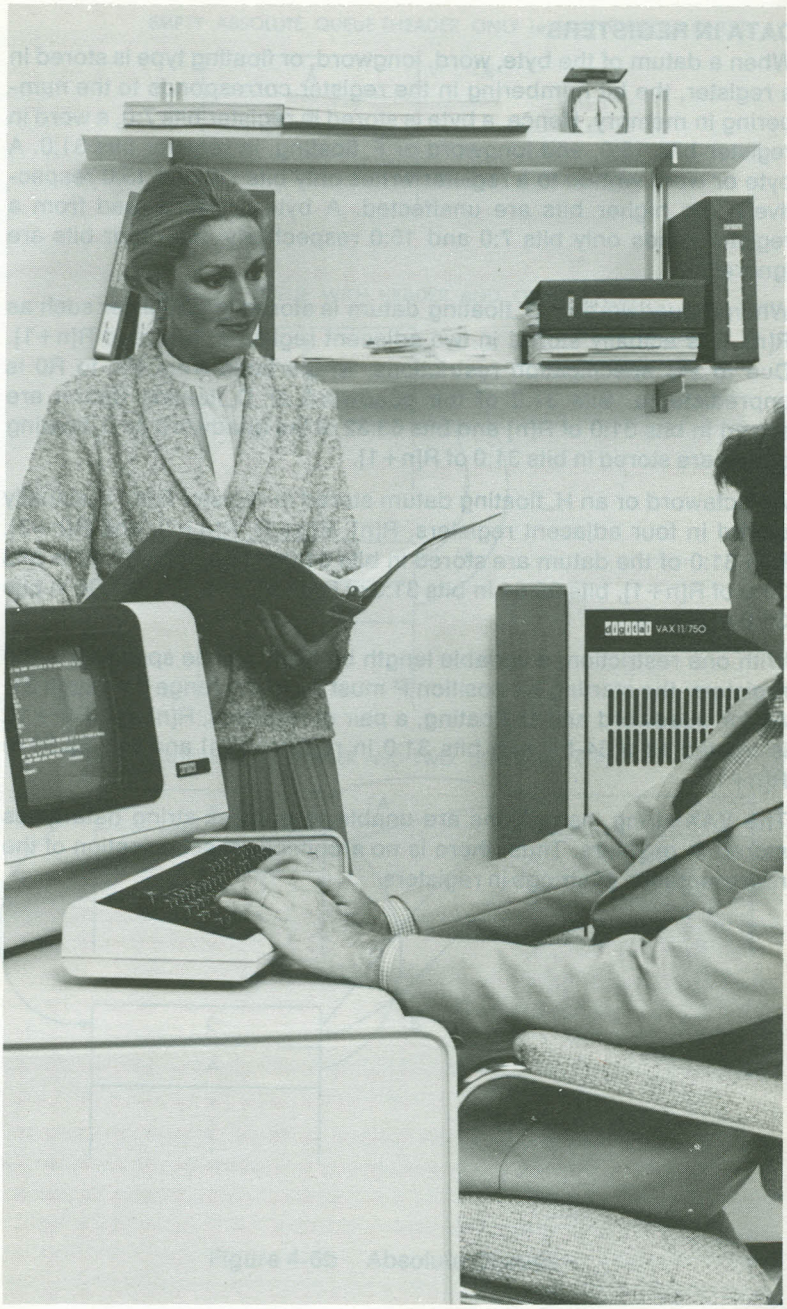
This Handbook is part of the VAX Handbook Set. It is the most deeply technical of the three Handbooks in the set, and should probably be read by people with some computer familiarity. In it you will get a thorough view of the instruction set, of memory management, of process structure, and of PDP-11 compatibility mode. A companion volume, the VAX Software Handbook, describes in more generic terms the VAX/VMS operating system, optional languages, software routines, and system services. And the VAX Hardware Handbook gives a complete view of the processors in the VAX family.

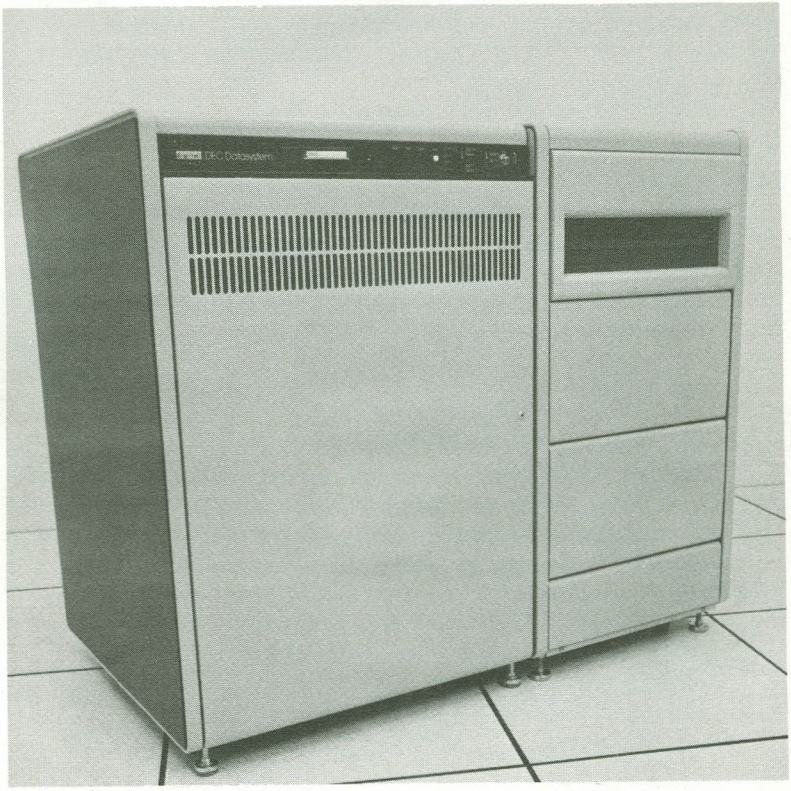
Note that this Handbook uses a consistent set of notational conventions. You will find their descriptions conveniently grouped together in Appendix A1.

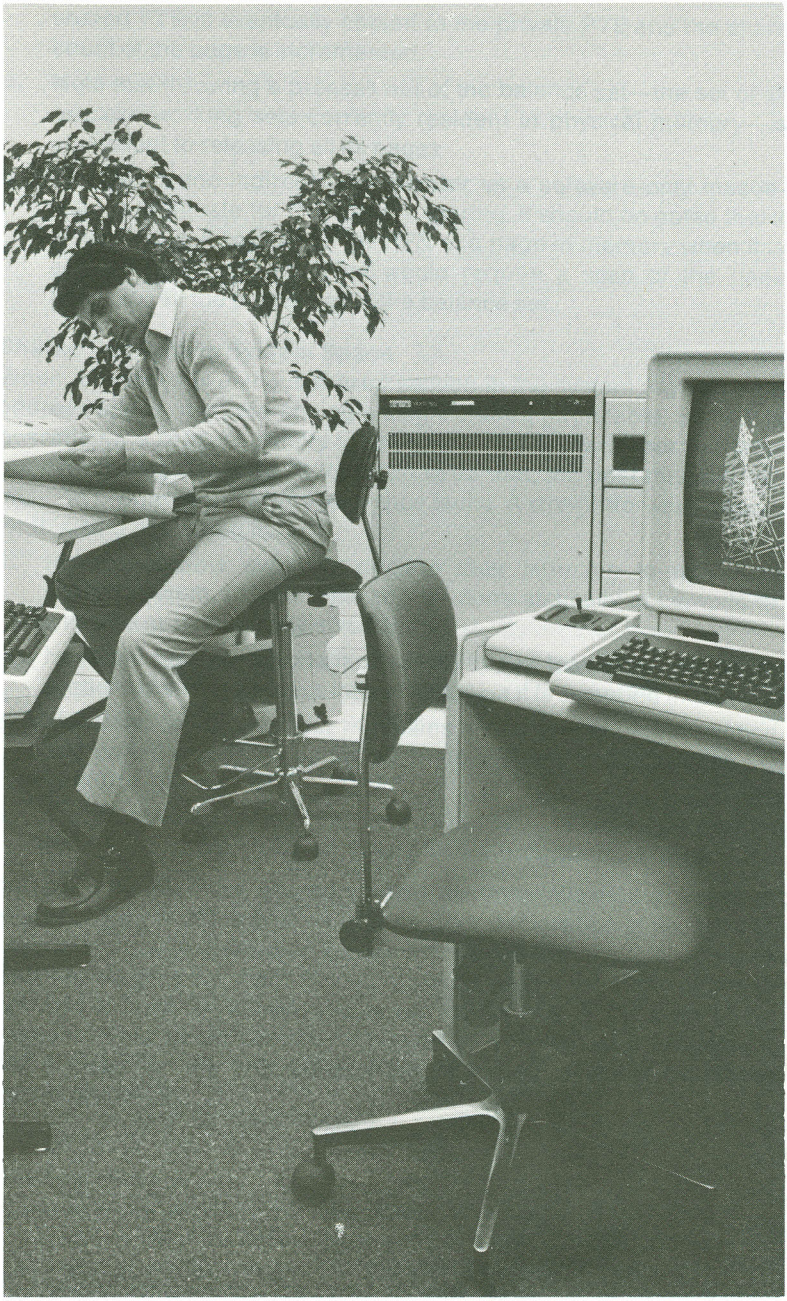
We hope that the Handbooks answer most of your questions about the VAX family, the computer architecture of the 1980s, and the huge selection of software available. If you have more questions, your DIGITAL Sales Representative will be happy to help you.



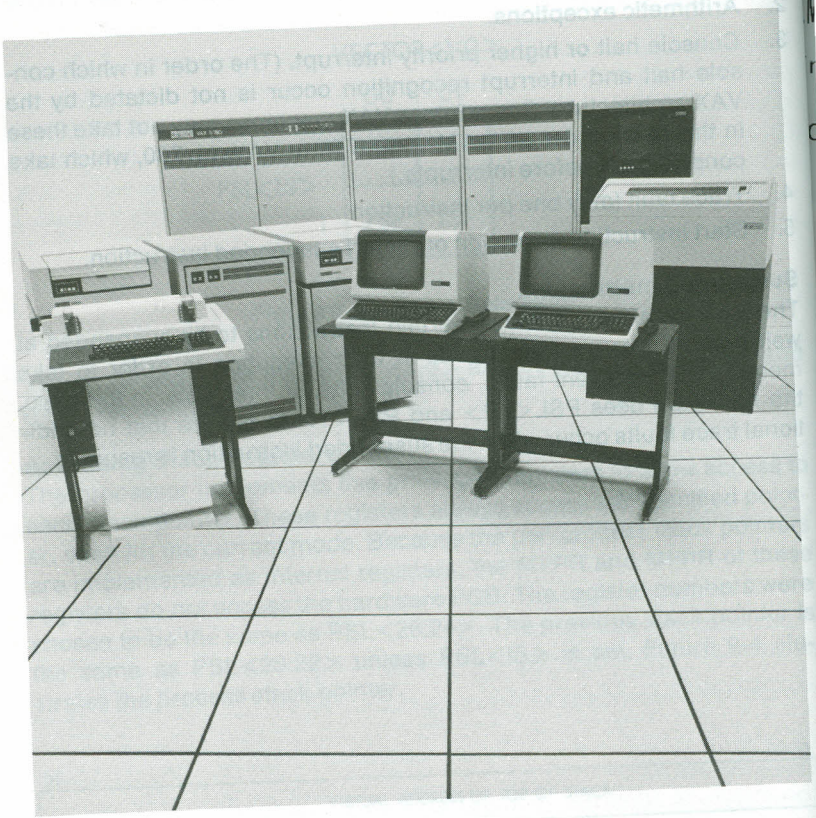



DIGITAL VAX 11/750

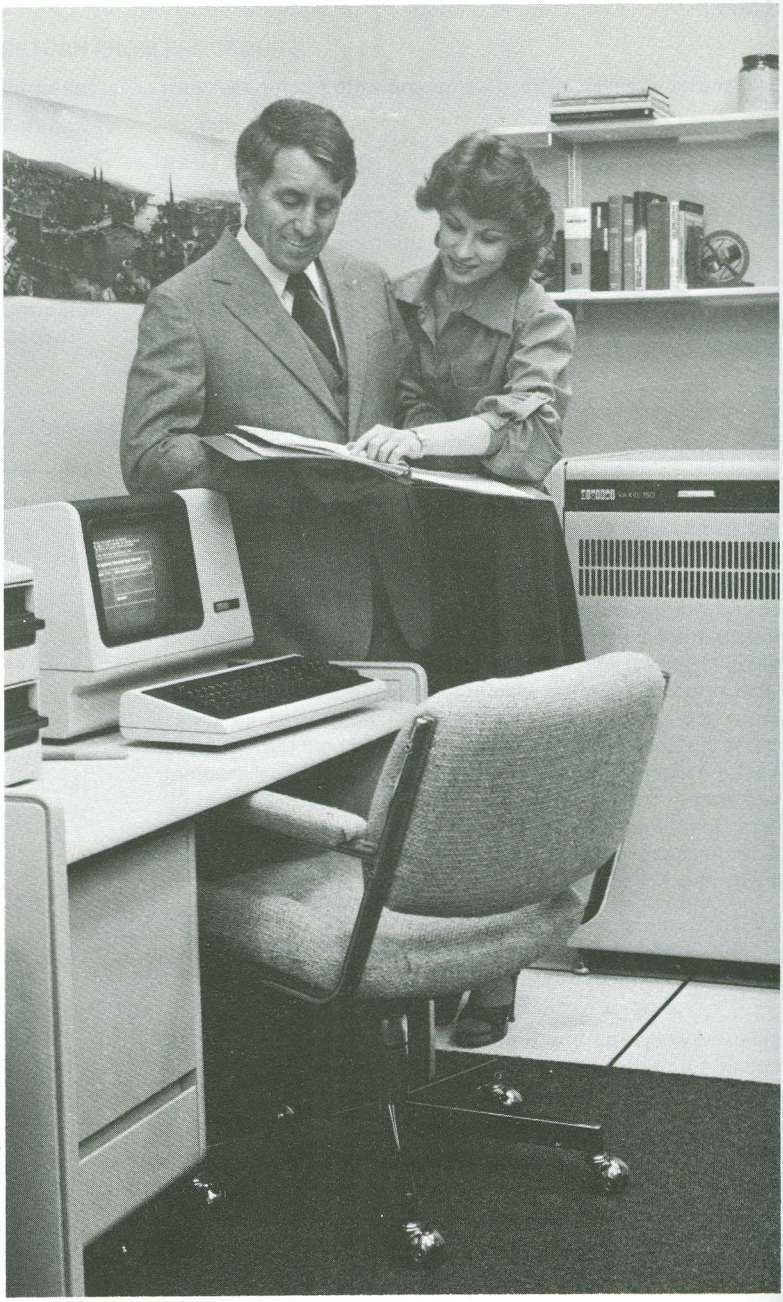

DEC Datasystem

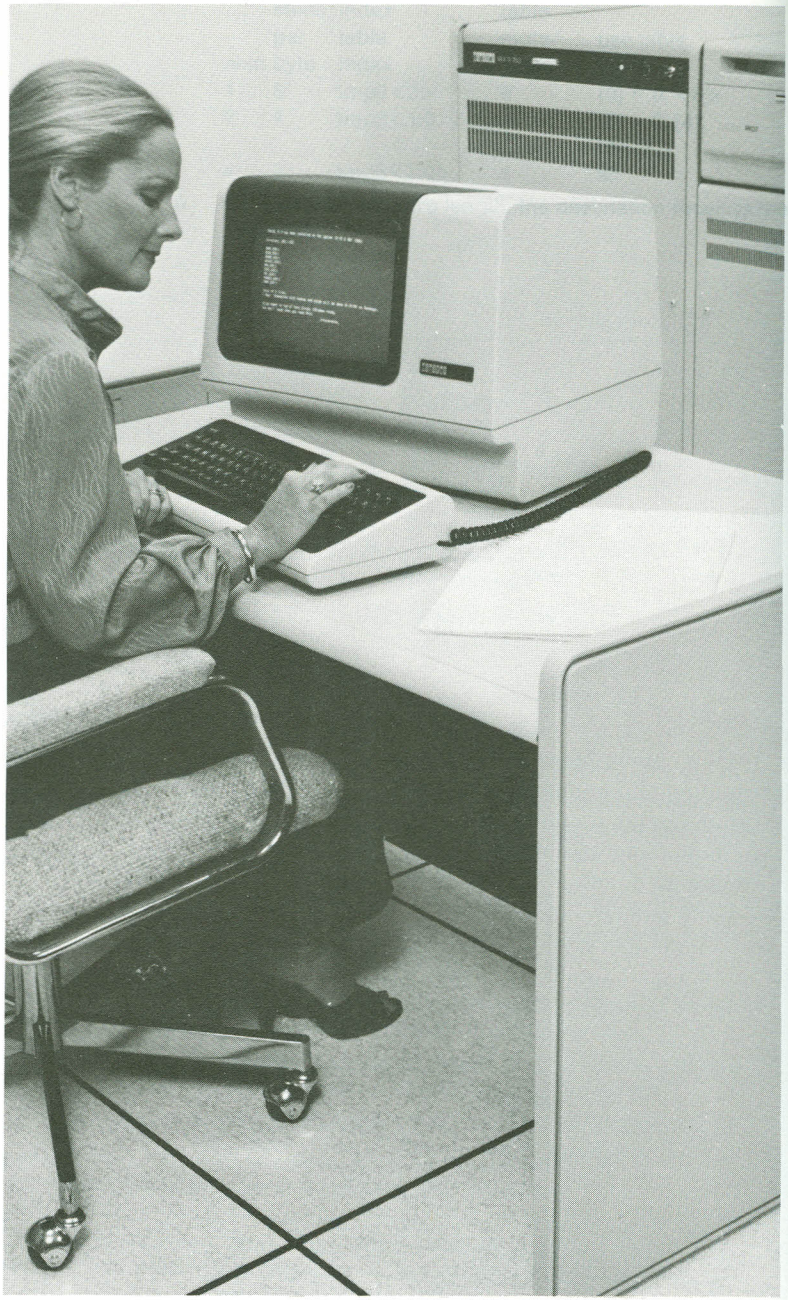


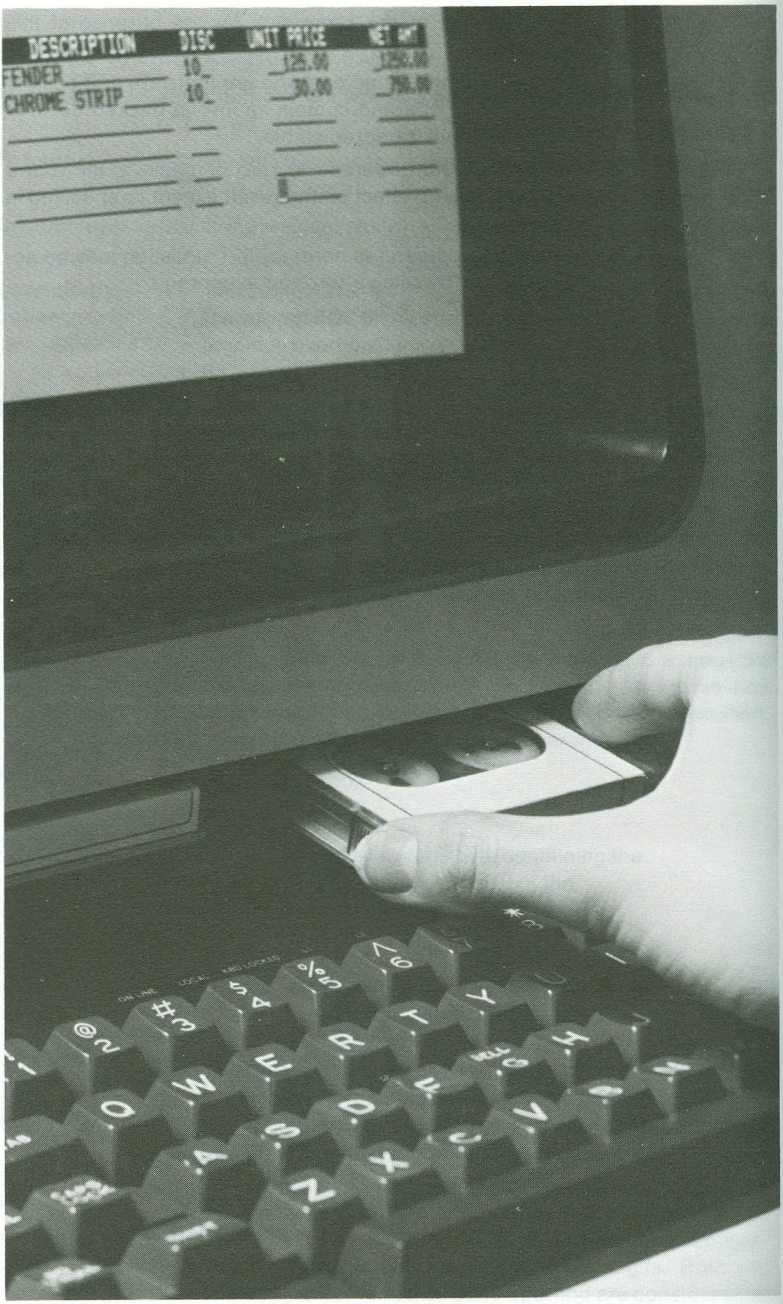

DIGITAL VAX 11/780

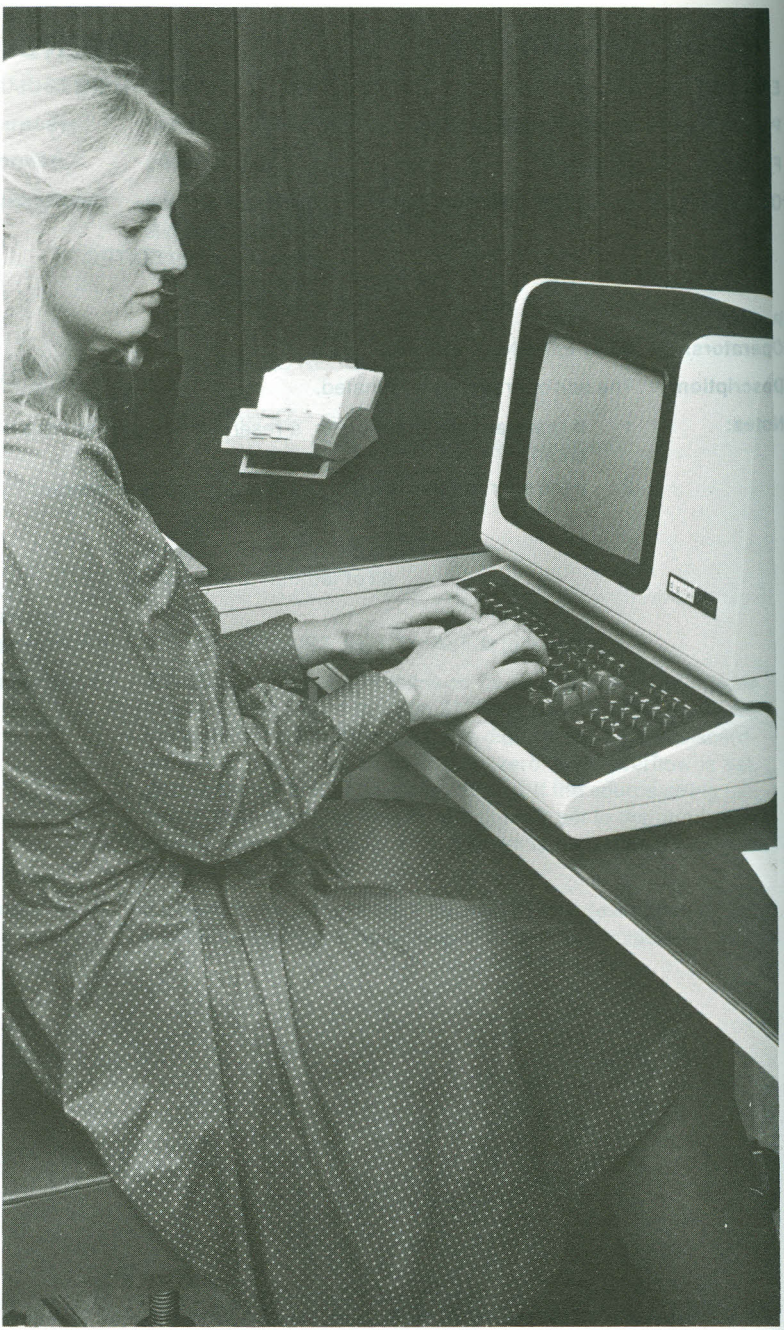



digital VAX 11/750






DESCRIPTION
DISC
UNIT PRICE
NET AMT
FENDER
10
125.00
1250.00
CHROME STRIP
10
30.00
750.00



digital

DIGITAL EQUIPMENT CORPORATION, Corporate Headquarters: Maynard, MA 01754, Tel. (617) 897-5111 — SALES AND SERVICE OFFICES; UNITED STATES — ALABAMA, Birmingham, Huntsville ARIZONA, Phoenix, Tucson ARKANSAS, Little Rock CALIFORNIA, Costa Mesa, El Segundo, Los Angeles, Oakland, Sacramento, San Diego, San Francisco, Monrovia, Santa Barbara, Santa Clara, Sherman Oaks COLORADO, Colorado Springs, Denver CONNECTICUT, Fairfield, Meriden DELAWARE, Newark FLORIDA, Miami, Orlando, Pensacola, Tampa GEORGIA, Atlanta HAWAII, Honolulu IDAHO, Boise ILLINOIS, Chicago, Peoria INDIANA, Indianapolis IOWA, Bettendorf KENTUCKY, Louisville LOUISIANA, New Orleans MARYLAND, Baltimore MASSACHUSETTS, Boston, Springfield, Waltham MICHIGAN, Detroit, Kalamazoo MINNESOTA, Minneapolis MISSOURI, Kansas City, St. Louis NEBRASKA, Omaha NEW HAMPSHIRE, Manchester NEW JERSEY, Cherry Hill, Parsippany, Princeton, Somerset NEW MEXICO, Albuquerque, Los Alamos NEW YORK, Albany, Buffalo, Long Island, New York City, Rochester, Syracuse, Westchester NORTH CAROLINA, Chapel Hill, Charlotte OHIO, Cincinnati, Cleveland, Columbus, Dayton OKLAHOMA, Tulsa OREGON, Portland PENNSYLVANIA, Harrisburg, Philadelphia, Pittsburgh RHODE ISLAND, Providence SOUTH CAROLINA, Columbia, Greenville TENNESSEE, Knoxville, Nashville TEXAS, Austin, Dallas, El Paso, Houston, San Antonio UTAH, Salt Lake City VERMONT, Burlington VIRGINIA, Fairfax, Richmond WASHINGTON, Seattle, Spokane WASHINGTON D.C. WEST VIRGINIA, Charleston WISCONSIN, Milwaukee INTERNATIONAL — EUROPEAN AREA HEADQUARTERS: Geneva, Tel: [41] (22)-93-33-11 INTERNATIONAL AREA HEADQUARTERS: Acton, MA 01754, U.S.A., Tel: (617) 263-6000 AUSTRALIA, Adelaide, Brisbane, Canberra, Hobart, Melbourne, Perth, Sydney, Townsville AUSTRIA, Vienna BELGIUM, Brussels BRAZIL, Rio de Janeiro, Sao Paulo CANADA, Calgary, Edmonton, Halifax, Kingston, London, Montreal, Ottawa, Quebec City, Regina, Toronto, Vancouver, Victoria, Winnipeg DENMARK, Copenhagen ENGLAND, Basingstoke, Birmingham, Bristol, Ealing, Epsom, Leeds, Leicester, London, Manchester, Reading, Welwyn FINLAND, Helsinki FRANCE, Bordeaux, Lyon, Paris, Puteaux, Strasbourg HOLLAND, Amstelveen, Delft, Utrecht HONG KONG IRELAND, Dublin ISRAEL, Tel Aviv ITALY, Milan, Rome, Turin JAPAN, Osaka, Tokyo MEXICO, Mexico City, Monterrey NEW ZEALAND, Auckland, Christchurch, Wellington NORTHERN IRELAND, Belfast NORWAY, Oslo, PUERTO RICO, San Juan SCOTLAND, Edinburgh REPUBLIC OF SINGAPORE SPAIN, Barcelona, Madrid SWEDEN, Gothenburg, Stockholm SWITZERLAND, Geneva, Zurich, WEST GERMANY, Berlin, Cologne, Frankfurt, Hamburg, Hannover, Munich, Nurnberg, Stuttgart