

PDP-8 Frequently Asked Questions

By Douglas Jones

From: jones@cs.uiowa.edu (Douglas W. Jones)
Newsgroups: alt.sys.pdp8
Subject: PDP-8 Frequently Asked Questions (posted every other month)
Date: 8 Jun 2001 08:08:08 GMT
Message-ID: <9fqtpj\$jde\$1@flood.weeg.uiowa.edu>
Summary: Answers to common questions about antique DEC PDP-8 computers.
Those posting to alt.sys.pdp8 should read this.
Keywords: FAQ DEC PDP 8

Archive-name: dec-faq/pdp8
Last-modified: Apr 8, 2001

Frequently Asked Questions about the DEC PDP-8 computer.

By Douglas Jones, jones@cs.uiowa.edu
(with help from many folks)

Reasonably current versions of this file is available by anonymous FTP
from:

<ftp://rtfm.mit.edu/pub/usenet/>
<ftp://ftp.uu.net/usenet/news.answers/dec-faq>
<ftp://src.doc.ic.ac.uk:/pub/usenet/news.answers/alt.sys.pdp8>

Reasonably current automatic translations of this document to HTML format
for the World Wide Web are available from:

<http://www.faqs.org/faqs/dec-faq/>
<http://www.cs.ruu.nl/wais/html/na-dir/dec-faq/.html>

An obsolete version of this file is available on the Walnut Creek USENET
FAQ CDROM; another version will be published as part of the FAQbook by
Pamela Greene et al.

This posting conforms to RFC1153 USENET digest format (with exceptions due
to the fact that it is not really a digest).

Contents:

- What is a PDP?
- What is a PDP-8?
- What is the PDP-8 instruction set?
- What does PDP-8 assembly language look like?
- What character sets does the PDP-8 support?
- What different PDP-8 models were made?
- What about the LINC-8 and PDP-12?

PDP-8 Frequently Asked Questions

By Douglas Jones

Where can I get a PDP-8 today?
Where can I get PDP-8 documentation?
What operating systems were written for the PDP-8?
What programming languages were supported on the PDP-8?
Where can I get PDP-8 software?
Where can I get additional information?
What use is a PDP-8 today?
Who's Who?

Subject: What is a PDP?

In 1957, Ken Olson and Harlan Anderson founded Digital Equipment Corporation (DEC), capitalized at \$100,000, and 70% owned by American Research and Development Corporation. Olson and Anderson had designed major parts of the AN/FSQ-7, the TX-0 and the TX-2 computers at Lincoln Labs. They wanted to call their company Digital Computer Corporation, but the venture capitalists insisted that they avoid the term Computer and hold off on building computers.

With facilities in an old woolen mill in Maynard Massachusetts, DEC's first product was a line of transistorized digital "systems modules" based on the modules used in building TX-2 at Lincoln Labs; these were plug-in circuit boards with a few logic gates per board. Starting in 1960, DEC finally began to sell computers (the formal acceptance of the first PDP-1 by BBN is reported in Computers and Automation, April 1961, page 8B). Soon after this, there were enough users that DECUS, the Digital Equipment Computer User's Society was founded.

DEC's first computer, the PDP-1, sold for only \$120,000 at a time when other computers sold for over \$1,000,000. (A good photo of a PDP-1 is printed in Computers and Automation, Dec. 1961, page 27). DEC quoted prices as low as \$85,000 for minimal models. The venture capitalist's insistence on avoiding the term computer was based on the stereotype that computers were big and expensive, needing a computer center and a large staff; by using the term Programmable Data Processor, or PDP, DEC avoided this stereotype. For over a decade, all digital computers sold by DEC were called PDPs. (In early DEC documentation, the plural form "PDPs" is used as a generic term for all DEC computers.)

In the early 1960's, DEC was the only manufacturer of large computers without a leasing plan. IBM, Burroughs, CDC and other computer manufacturers leased most of their machines, and many machines were never offered for outright sale. DEC's cash sales approach led to the growth of third party computer leasing companies such as DELOS, a spinoff of BB&N.

PDP-8 Frequently Asked Questions

By Douglas Jones

DEC built a number of different computers under the PDP label, with a huge range of price and performance. The largest of these are fully worthy of large computer centers with big support staffs. Some early DEC computers were not really built by DEC. With the PDP-3 and LINC, for example, customers built the machines using DEC parts and facilities. Here is the list of PDP computers:

| MODEL | DATE | PRICE | BITS | NUMBER | COMMENTS |
|--------|-------|-----------|-------|----------|--|
| ===== | ===== | ===== | ===== | ===== | ===== |
| PDP-1 | 1960 | \$120,000 | 18 | 50 | DEC's first computer |
| PDP-2 | | NA | 24 | | - Never built? Prototype only? |
| PDP-3 | | NA | 36 | | One built by a customer*, not by DEC. |
| PDP-4 | 1962 | \$60,000 | 18 | 45 | Predecessor of the PDP-7. |
| PDP-5 | 1963 | \$27,000 | 12 | 1,000 | The ancestor of the PDP-8. |
| PDP-6 | 1964 | \$300,000 | 36 | 23 | A big computer; 23 built, most for MIT. |
| PDP-7 | 1965 | \$72,000 | 18 | 120 | Widely used for real-time control. |
| PDP-8 | 1965 | \$18,500 | 12 | ~50,000 | The smallest and least expensive PDP. |
| PDP-9 | 1966 | \$35,000 | 18 | 445 | An upgrade of the PDP-7. |
| PDP-10 | 1967 | \$110,000 | 36 | **~700 | A PDP-6 followup, great for timesharing. |
| PDP-11 | 1970 | \$10,800 | 16 | >600,000 | DEC's first and only 16 bit computer. |
| PDP-12 | 1969 | \$27,900 | 12 | 725 | A PDP-8 relative. |
| PDP-13 | | NA | | | - Bad luck, there was no such machine. |
| PDP-14 | | | | *** | A ROM-based programmable controller. |
| PDP-15 | 1970 | \$16,500 | 18 | 790 | A TTL upgrade of the PDP-9. |
| PDP-16 | 1972 | NA | 8/16 | ? | A register-transfer module system. |

* Scientific Engineering Institute of Waltham MA. SEI was allegedly founded in 1956 by the CIA to study the effects of microwaves (radar) on the human brain. If so, the PDP-3 may have been used as an instrumentation computer. More info on the CIA connection and the use of the PDP-3 would be nice!

** Includes DECSYSTEM 20.

Corrections and additions to this list are welcome! The prices given are for minimal systems in the year the machine was first introduced. Most of the production run numbers come from "Computer Engineering" by Bell, Mudge and McNamara, 1978, or from Computers and Automation's computer census figures published regularly throughout the 1960's. The bits column in the table indicates the word size. Note that the DEC PDP-10 became the DECSYSTEM-20 as a result of marketing considerations, and DEC's VAX series of machines began as the Virtual Address eXtension of the never-produced PDP-11/78.

It is worth mentioning that it is widely (but somewhat incorrectly)

PDP-8 Frequently Asked Questions

By Douglas Jones

accepted that the Data General Nova (see photo, Computers and Automation, Nov. 1968, page 48) grew out of the PDP-X, a 16-bit multi-register version of the PDP-8 designed by Edson DeCastro, Henry Burkhardt and Dick Soggee. (DeCastro was one of DEC's key design engineers; his name appears on many of the blueprints for machines from the PDP-5 up through the PDP-8/L).

A prototype PDP-X was built at DEC; this and a competing 16-bit design were apparently submitted to Harold McFarland at Carnegie-Mellon University for evaluation; McFarland (and perhaps Gordon Bell, who was at C-MU at the time) evaluated the competing designs and rejected both in favor of what we now know as the PDP-11. (I was at Carnegie-Mellon at the time, and McFarland gave a guest lecture in a class I attended telling part of this story.) Some speculate, incorrectly, that Bell rejected the Nova design because the competing proposal used the register-transfer notation he had introduced in "Bell and Newell, Computer Structures -- Readings and Examples". An alternate and equally unfounded story is that the reason DEC never produced a PDP-13 was because the number 13 had been assigned to what became the Nova.

In any case, when DeCastro, Burkhardt and Soggee founded Data General, Ken Olson at DEC was very angry, claiming for a long time that the Nova design was stolen. Gordon Bell and others concluded that the Nova design was sufficiently original that a lawsuit was unwarranted, but the feud between DeCastro and Olson lasted until after Ken Olson left DEC. It is more correct to say that the Nova is a reaction to the PDP-X than to say that it is based on the PDP-X. I am indebted to Jim Campbell, retired VP at Data General, for some of the details of this story.

Today, all of the PDP machines are in DEC's corporate past, except the PDP-11 family, which survives as a line of microcomputers; DEC has promised to discontinue PDP-11 sales on Sept. 30, 1996. Occasionally, some lab has built a machine out of DEC hardware and called it a PDP with a new number. For example, the Australian Atomic Energy Commission once upgraded a PDP-7 by adding a PDP-15 on the side; they called the result a PDP-22. There is also a story about the PDP-2 1/2, built by Ed Rawson of the American Science Institute out of surplus modules that were originally used in the prototype PDP-2.

In 1998, Compaq purchased DEC, and it is unclear how long DEC will retain any semblance of its original identity as a division of a larger company.

Subject: What is a PDP-8?

PDP-8 Frequently Asked Questions

By Douglas Jones

The PDP-8 family of minicomputers were built by Digital Equipment Corporation between 1965 and 1990, although it is worth noting that the term minicomputer first came into prominence after the machine was introduced. The first use of the term appears to have been made by the head of DEC's operations in England, John Leng. He sent back a sales report that started: "Here is the latest minicomputer activity in the land of miniskirts as I drive around in my [Austin] Mini Minor." The term quickly became part of DEC's internal jargon and spread from there; the first computer explicitly sold as a minicomputer, though, was made by Interdata (See the Interdata ad in Computers and Automation, May 1968, page 10).

The PDP-8 was largely upward compatible with the PDP-5, a machine that was unveiled on August 11, 1963 at WESCON, and the inspiration for that machine came from two earlier machines, the LINC and the CDC 160. All of these machines were characterized by a 12 bit word with little or no hardware byte structure, typically 4K words of memory, and simple but powerful instruction sets.

Although some people consider the CDC 160 the first minicomputer, the PDP-8 was the definitive minicomputer. By late 1973, the PDP-8 family was the best selling computer in the world, and it is likely that it was only displaced from this honor by the Apple II (which was displaced by the IBM PC). Most models of the PDP-8 set new records as the least expensive computer on the market at the time of their introduction. The PDP-8 has been described as the model-T of the computer industry because it was the first computer to be mass produced at a cost that just about anyone could afford.

C. Gordon Bell has said that the basic idea of the PDP-8 was not really original with him. He gives credit to Seymour Cray (of CDC and later Cray) for the idea of a single-accumulator 12 bit minicomputer. Cray's CDC 160 family (see CACM, march 1961, photo on page 244, text on page 246) was such a machine, and in addition to the hundreds of CDC 160 systems sold as stand-alone machines, a derivative 12 bit architecture was used for the I/O processors on Cray's first great supercomputer, the CDC 6600.

Note that Cray's 12 bit machines had 6 basic addressing modes with variable length instruction words and other features that were far from the simple elegance of the PDP-8. Despite its many modes, the CDC 160 architecture lacked the notion of current page addressing, it had no unconditional jump instruction, and the I/O instructions all blocked the CPU until I/O complete. As a result, the PDP-8 is both far more flexible and it supports much tighter programming styles.

PDP-8 Frequently Asked Questions

By Douglas Jones

The teleprinter flag is set by the completion of the TPC operation (as a result, on startup, many applications output a null in order to get things going). TCF clears the flag, and TPC outputs the low order 8 bits of the accumulator. The commonly used TLS instruction is the or of TCF and TPC. To output a character, first use TSF to poll the flag, then write the character with TLS.

IOT instructions may be used to initiate data break transfers from block devices such as disk or tape. The term "data break" was, for years, DEC's preferred term for cycle-stealing direct-memory-access data transfers.

Some CPU functions are accessed only by IOT instructions. For example, interrupt enable and disable are IOT instructions:

```

|1|1|0|_|_|_|_|_|_|_|_|_|
|0 0 0 0 0 0|0 0 1 - ION - interrupts turn on
|0 0 0 0 0 0|0 1 0 - IOF - interrupts turn off

```

An interrupt is requested when any device raised its flag. The console master clear switch resets all flags and disables interrupts. In effect, an interrupt is a JMS instruction to location zero, with the side effect of disabling interrupts. The interrupt service routine is expected to test the device flags and perform the operations needed to reset them, and then return using ION immediately before the indirect return JMP. The effect of ION is delayed so that interrupts are not enabled until after the JMP.

The instructions controlling the optional memory management unit are also IOT instructions. This unit allows the program to address up to 32K of main memory by adding a 3 bit extension to the memory address. Two extensions are available, one for instruction fetch and direct addressing, the other for indirect addressing.

A wide variety of operations are available through the OPR microcoded instructions:

```

Group 1 |1|1|1|0|_|_|_|_|_|_|_|_|_|
          1 - CLA - clear AC
          1 - CLL - clear the L bit
          1 - CMA - ones complement AC
          1 - CML - complement L bit
          1 - IAC - increment <L,AC>
          1 0 0 - RAR - rotate <L,AC> right
          0 1 0 - RAL - rotate <L,AC> left
          1 0 1 - RTR - rotate <L,AC> right twice

```


PDP-8 Frequently Asked Questions

By Douglas Jones

Subject: What does PDP-8 assembly language look like?

There are many different assemblers for the PDP-8, but most use a compatible basic syntax; here is an example:

```
START, CLA CLL      / Clear everything
      TAD      X      / Load X
      AND I   Y      / And with the value pointed to by Y
      DCA      X      / Store in X
      HLT
X,      1           / A variable
Y,      7           / A pointer
```

Note that labels are terminated by a comma, and comments are separated from the code by a slash. There are no fixed fields or column restrictions. The "CLA CLL" instruction on the first line is an example of the microcoding of two of the Group 1 operate instructions. CLA alone has the code 7200 (octal), while CLL has the code 7100; combining these as "CLA CLL" produces 7300. As a general rule, except when memory reference instructions are involved, the assembler simply ors together the values of all blank separated fields between the label and comment.

Indirection is indicated by the special symbol I in the operand field, as in the third line of the example. The typical PDP-8 assembler has no explicit notation to distinguish between page zero and current page addresses. Instead, the assembler is expected to note the page holding the operand and automatically generate the appropriate mode. If the operand is neither in the current page nor page zero, some assemblers will raise an error, others will automatically generate an indirect pointer to the off-page operand; this should be avoided because it only works for directly addressed off-page operands, and only when the memory management unit is not being used to address a data field other than the current instruction field.

Note, in the final two lines of the example, that there is no "define constant" pseudo-operation. Instead, where a constant is to be assembled into memory, the constant takes the place of the op-code field.

The PDP-8 has no immediate addressing mode, but most assemblers provide a notation to allow the programmer to ignore this lack:

```
TAD      (3)      / add 3, from memory on the current page.
TAD      [5]      / add 5, from memory on page zero.
JMP I   (LAB)    / jump indirect through the address of LAB.
```

Assemblers that support this automatically fill the end of each page

PDP-8 Frequently Asked Questions

By Douglas Jones

with constants defined in this way that have been accumulated during the assembly of that page. Note that the variants "(3" and "[5" (with no closing parentheses) are usually allowed but the use of this sloppy form is discouraged. Furthermore, the widely used PAL8 assembler interprets the unlikely operand "(3)+1" as being the same as "(3+1)".

Arithmetic is allowed in operand fields and constant definitions, with expressions evaluated in strict left-to-right order, as:

```
TAD    X+1    / add the contents of the location after X.
TAD    (X-1)  / add the address of the location before X.
```

Other operators allowed include and (&), or (!), multiply (^) and divide (%), as well as a unary sign (+ or -). Unfortunately, one of the most widely used assemblers, PAL8, has trouble when unary operators are mixed with multiplication or division.

Generally, only the first 6 characters of identifiers are significant and numeric constants are evaluated in octal.

Other assembly language features are illustrated below:

```
/ Comments may stand on lines by themselves
/ Blank lines are allowed

*200          / Set the assembly origin to 200 (octal)

NL0002=CLA CLL CML RTL      / Define new opcode NL0002.

NL0002       / Use new opcode (load 0002 in AC)
JMP    .-1   / Jump to the previous instruction

X1=    10    / Define X1 (an auto-index register address)
LETA=  "A    / Define LETA as 000011000001 (ASCII A)

TAD I  X1    / Use autoindex register 1

IAC; RAL    / Multiple instructions on one line

$          / End of assembly
```

The assembly file ends with a line containing a \$ (dollar sign) not in a comment field.

The \$, * and = syntax used by most PDP-8 assemblers replaces functions performed by pseudo-operations on many other assemblers. In addition, PAL8, the most widely used PDP-8 assembler supports the following

PDP-8 Frequently Asked Questions

By Douglas Jones

pseudo-operations:

| | | |
|---------|---------|--|
| DECIMAL | | / Interpret numeric constants in base 10 |
| OCTAL | | / Interpret numeric constants in base 8 |
| EJECT | | / Force a page eject in the listing |
| XLIST | | / Toggle listing |
| XLIST | N | / Turn on listing if N=0, off if N=1 |
| PAGE | | / Advance location counter to next page |
| PAGE | N | / Set location counter start of page N |
| FIELD | N | / Assemble into extended memory field N |
| TEXT | "STR" | / Pack STR into consecutive 6 bit bytes |
| ZBLOCK | N | / Allocate N words, initialized to zero |
| IFDEF | S <C> | / Assemble C if symbol S is defined |
| IFNDEF | S <C> | / Assemble C if symbol S is not defined |
| IFZERO | E <C> | / Assemble C if expression E is zero |
| IFNZRO | E <C> | / Assemble C if expression E is not zero |
| FIXMRI | OP= VAL | / Define OP as memory reference |

instruction

Conditionally assembled code must be enclosed in angle brackets. The enclosed code may extend over multiple lines and, because different assemblers treat comments within conditionals differently, the closing bracket should not be in a comment and any brackets in comments should be balanced.

Subject: What character sets does the PDP-8 support?

From the beginning, PDP-8 software has generally assumed that textual I/O would be in 7 bit ASCII. Most early PDP-8 systems used teletypes as console terminals; as sold by DEC, these were configured for mark parity, so most older software assumes 7 bit ASCII, upper case only, with the 8th bit set to 1. On output, lines are generally terminated with both CR and LF; on input, CR is typically (but not always) the line terminator and LF is typically ignored. In addition, the tab character (HT) is generally allowed, but software support output of text containing tabs varies.

One difficulty with much PDP-8 software is that it bypasses the device handlers provided by the operating system and goes directly to the device. This results in very irregular device support, so that, for example, control-S and control-Q work to start and stop output under OS/8, but the OS/8 PAL assembler ignores them when reporting errors.

Most of the better engineered PDP-8 software tends to fold upper and lower case on input, and it ignores the setting of the 8th bit. Older

PDP-8 Frequently Asked Questions

By Douglas Jones

PDP-8 software will generally fail when presented with lower case textual input (this includes essentially all OS/8 products prior to OS/278 V1).

Internally, PDP-8 programmers are free to use other character sets, but the "X notation provided by the assembler encourages use of 7 bit ASCII with the 8th bit set to 1, and the TEXT pseudo-operation encourages the 6 bit character set called "stripped ASCII". To map from upper-case-only ASCII to stripped ASCII, each 8 bit character is anded with octal 77 and then packed 2 characters per word, left to right. Many programs use a semi-standard scheme for packing mixed upper and lower case into 6 bit TEXT form; this uses ^ to flip from upper to lower case or lower to upper case, % to encode CR-LF pairs, and @ (octal 00) to mark end of string. Note that this scheme makes no provision for encoding the %, ^ and @ characters, nor does it allow control characters other than the CR-LF pair.

The P?S/8 operating system supports a similar 6 bit text file format, where upper and lower case are folded together, tabs are stored as _ (underline), end-of-line is represented by 00, padded with any nonzero filler to a word boundary, and end of file is 0000.

Files under the widely used OS/8 system consist of sequences of 256 word blocks. When used for text, each block holds 384 bytes, packed 3 bytes per pair of words as follows:

```
aaaaaaa          ccccaaaaaaa
bbbbbbb          CCCCbbbbbbb
ccccCCCC
```

Control Z is used as an end of file marker. Because most of the PDP-8 system software was originally developed for paper tape, binary object code is typically stored in paper-tape image form using the above packing scheme.

Subject: What different PDP-8 models were made?

The total sales figure for the PDP-8 family is estimated at over 300,000 machines. Over 7000 of these were sold prior to 1970, and 30,000 were sold by 1976. During the PDP-8 production run, a number of models were made, as listed in the following table. Of these, the PDP-8/E is generally considered to be the definitive machine. If the PDP-8 is considered to be the Model T of the computer industry, perhaps the PDP-8/E should be considered to be the industry's Model A.

PDP-8 Frequently Asked Questions

By Douglas Jones

| MODEL | DATES | SALES | COST | TECHNOLOGY | REMARKS |
|-------------------|--------|-------|----------|-----------------|--------------------|
| PDP-5 | 63-67 | 116 | | Transistor | |
| PDP-8 | 65-69 | 1450 | \$18,500 | Transistor | |
| LINC-8 | 66-69 | 142 | \$38,500 | Transistor | |
| PDP-8/S | 66-70 | 1024 | \$10,000 | Transistor | Very slow |
| PDP-8/I | 68-71 | 3698 | \$12,800 | TTL | |
| PDP-8/L | 68-71 | 3902 | \$8,500 | TTL | Scaled down 8/I |
| PDP-12 | 69-73? | 3500? | \$27,900 | TTL | Followup to LINC-8 |
| PDP-8/E | 70-78 | >10K? | \$6,500 | TTL MSI Omnibus | |
| PDP-8/F | 72-78? | >10K? | <\$5K | TTL MSI Omnibus | Based on 8/E CPU |
| PDP-8/M | 72-78? | >10K? | <\$5K | TTL MSI Omnibus | OEM version of |
| 8/F | | | | | |
| PDP-8/A | 75-84? | >10K? | \$1,317 | TTL LSI Omnibus | New CPU or 8/E |
| CPU | | | | | |
| VT78 | 78-80 | | \$7,995 | Intersil 6100 | Workstation |
| DECmate I | 80-84 | | | Harris 6120 | Workstation |
| DECmate II | 82-86 | | \$1,435 | Harris 6120 | Workstation |
| DECmate III | 84-90 | | \$2,695 | Harris 6120 | Workstation |
| DECmate III+85-90 | | | | Harris 6120 | Workstation |

Additional information is available in part two of this FAQ, where all known models of the PDP-8, along with variants, alternate marketing names, and other peculiarities are given.

The last years of the PDP-8 family were dominated by the PDP-8 compatible microprocessor based VT78 and DECmate workstations. The Intersil 6100, also known as the CMOS-8 chip, was developed in 1976; GE later acquired Intersil. DEC also used the followup Harris 6120 microprocessors (Introduced 1981) in many peripheral controllers for the PDP-11 and PDP-15 as well as in the DECmate series of systems. While all of the earlier PDP-8 systems were open architecture systems, the DECmates had closed architectures with an integrated console terminals and limited peripheral options. It is interesting to note that the Harris 6120 was a 10Mhz chip and some chips could be clocked at 15Mhz; furthermore, the 6120 was essentially based on gate array technology.

The following PDP-8 compatible or semi-compatible machines were made and sold by others; very little is known about many of these:

| MODEL | DATE | MAKER, NOTES |
|-----------|------|--|
| TPA1001 | 69 | Hungarian, KFKI product, transistorized. |
| TPA1001/i | 71 | Hungarian, KFKI, IC version of 1001. |
| TPA/i | 71 | Hungarian, KFKI, renamed TPA1001/i |
| TPA/l | 7? | Hungarian, KFKI, enhanced TPA/i. |
| TPA1/128H | 7? | Hungarian, KFKI, TPA/l with 128K memory. |

PDP-8 Frequently Asked Questions

By Douglas Jones

| | | |
|----------------------|-------|---|
| TPA/s | 7? | Hungarian, KFKI, based on Intersil CPU chip. |
| TPA Quadro | 8? | Hungarian, KFKI, comparable to a DECmate. |
| Electronica-100 | ? | Russian, discrete transistor technology. |
| Electronica-100I | ? | Russian, probably a PDP-8/I clone. |
| Electrotechnica-100I | ? | Yugoslavian, PDP-8/I? Possibly same as above. |
| Saratov-2 | ? | Russian, built like a PDP-8/M but bulkier. |
| SPEAR u-LINC 100 | ? | SPEAR, Inc, Waltham Mass (a LINC clone!) |
| SPEAR u-LINC 300 | ? | SPEAR, Inc, Waltham Mass (a LINC clone!) |
| DCC-112 | 70 | Digital Computer Controls, PDP-8/L clone. |
| DCC-112H | 71 | Digital Computer Controls |
| MPS-1 | 74 | Fabritek, PDP-8/L clone |
| MP-12 | 74 | (is this just different numbering of above?) |
| 6100 Sampler | 76? | Intersil, their IM6100 promotional kit |
| Intercept I | 7? | Intersil, based on IM6100 |
| Intercept Jr | 7? | Intersil, based on IM6100 |
| TLF MINI-12 | 77 | Based on IM6100, in an elegant package. |
| PCM-12 | 7? | Pacific CyberMetrix, based on Intercept bus |
| PCM-12A | 77 | Pacific CyberMetrix, fixed to clock at 4MHz |
| SBC-8 | 84-88 | CESI, Based on IM6120? SCSI bus |

More information on the Hungarian TPA series, built by KFKI (the Central Research Institute for Physics), was provided by Varga Akos Endre, hamster@telnet.hu; information on and photos of these machines are currently available from:

http://www.internetto.hu/muzeum/e_tpa.html
http://www.internetto.hu/muzeum/e_tpai.html

The original machine in this series, the TPA1001, was built from the description in DEC's Small Computer Handbook. Only after the series was in production, when a machine was exhibited in Ljubljana, Yugoslavia, was full DEC compatibility demonstrated, when a DEC user booted the TPA machine from a DEC paper tape. By the end of the TPA production run, around 900 PDP-8 compatible machines had been built. Given the Soviet era central planning for the computer industries in eastern Europe, it is quite possible that the Electrotechnica and Electronica models listed above may have been TPA machines packaged for use in the USSR and other Soviet Block countries.

It is amusing to note that the name TPA is very similar in origin to the name PDP used by DEC! There was a decree that computer development in Hungary was to cease, with all computers to be purchased from the USSR. In response, the

PDP-8 Frequently Asked Questions

By Douglas Jones

people at KFKI ceased developing computers and began developing "Stored Program Analyzers" or, the acronym for which is TPA in Hungarian.

The CESI (Computer Extension Systems, Inc.) machine had 128K words of local RAM on each processor card and allowed up to 4 processor cards per OMNIBUS, along with 128K words of global shared memory. 3 AMD 2901 bit-slice processor chips were used to build the 12-bit ALU and data paths, controlled by an 80-bit microword.

Subject: What about the LINC/8 and PDP-12?

Wesley Clark and Charles Molnar, then at Lincoln Labs, built the LINC, or Laboratory INstrumentation Computer, as a personal laboratory computer, finishing the first in March 1962. The machine was developed in response to the needs of Mary Brazier, a neurophysiologist at MIT who needed better laboratory tools, and it was a followup to the Average Response Computer, an 18 bit special purpose machine built in 1958 for the same purpose. When Lincoln Labs decided that the LINC did not fit their mission, in January 1963, the project moved to MIT, and then in 1964, to Washington University in St Louis. The National Institute of Health funded the project as an experiment to see if computers would be a productive tool in the life sciences. By the end of 1963, 20 LINC's had been built and debugged, many by their eventual users.

Over 24 LINC systems were built by customers before late 1964 when DEC began selling a commercial version (see Computers and Automation, Nov. 1964, page 43). By the time DEC introduced the LINC-8, 43 LINC systems had been installed (see Computers and Automation, Mar. 1966, page 34). In total, 50 LINC systems were built, 21 by DEC, 29 by customers (see Digital at Work, page 52). A photo of the last LINC in production use is available from <http://www.mit.edu:8001/people/ijs/epl/LINC.html> Wesley Clark wrote a history of the LINC, "The LINC was Early and Small", published in "A History of Personal Workstations," ACM Press, 1988, page 347.

The LINC was the first 12 bit minicomputer built using DEC hardware. Like the PDP-5 and other early DEC computers, it was built with System Modules, DEC's first family of logic modules. Along with the CDC 160, it paved the way for the PDP-5 and PDP-8.

When compared with the PDP-8, the LINC instruction set was not as well suited for general purpose computation, but the common peripherals needed for lab work such as analog-to-digital and digital-to-analog

PDP-8 Frequently Asked Questions

By Douglas Jones

converters were all bundled into the LINC system. Users judged it to be a superb laboratory instrument.

One of the major innovations introduced with the LINC was the LINCtape, designed by Tom Stockebrand scaled down from an experimental tape drive developed for the TX-2 at Lincoln Labs. LINCtapes could be carelessly pocketed or dropped on the floor without fear of data loss, and they allowed random access to data blocks. Stockebrand improved on this idea slightly after he came to DEC, where the improved idea was called DECTape; DECTape was widely used with all DEC computers made in the late 1960's and early 1970's.

The motives behind the development of LINCtape were the same motives that led IBM to develop the floppy disk almost a decade later, and in fact, DECTape survived as a widely used medium until DEC introduced the RX01 8 inch floppy disk drive around 1975, and even after this, DECTape was only slowly phased out.

Within a year of the introduction of the PDP-8, DEC released the LINC-8, a machine that combined a PDP-8 with a LINC in one package. The success of the LINC-8 led DEC to re-engineer the machine using TTL logic in the late 1960's; the new version was originally to be called the LINC-8/I, but it was sold as the PDP-12. Both the LINC-8 and the PDP-12 had impressive consoles, with separate sets of lights and switches for the LINC and PDP-8 halves.

The success of the LINC-8 also led to the development of a clone, the SPEAR micro-LINC. This machine used Motorola MECL integrated circuits and was available for delivery in (June 1965? this date must be wrong!).

The LINC-8 and PDP-12 could run essentially any PDP-8 or LINC program, with the exception of the few programs that relied on the primitive interrupt structure of the original LINC architecture; on the LINC-8, all interrupts were handled by the PDP-8 side of the hardware. Because the LINC-8 and PDP-12 had instructions for switching between modes, a new body of software was developed that required both modes.

One feature of LINC and LINC-8 software is the common use of the graphic display for input-output. These machines were some of the first to include such a display as a standard component, and many programs used the knobs on the analog to digital converter to move a cursor on the display in the way we now use a mouse.

Various versions of LAP, the Linc Assembly Program, were the dominant assemblers used on the LINC; the original version of LAP was a cross assembler written on the TX-2. WISAL (WISconson Assembly Language) or LAP6-W was the version of this assembler that survived to run on the

PDP-8 Frequently Asked Questions

By Douglas Jones

PDP-12. Curiously, this includes a PDP-8 assembler written in LINC code.

LAP6-DIAL (Display Interactive Assembly Language) evolved from this on the PDP-12 to become the dominant operating system for the PDP-12. The 8K version of this is DIAL MS (Mass Storage), even if it has only two LINCtape drives. These were eventually displaced by the OS/8 variant known as OS/12.

Subject: Where can I get a PDP-8 today?

The IM6100 chip may still be available (Electronic Expeditors, (818)781-1910

(in North America) listed them at US\$23.50 each as of 10/1994), and CESI may still make their clone, for a high price, but you can't buy a new DEC PDP-8. There are quite a few PDP-8 machines to be found in odd places on the used equipment market. They were widely incorporated into products such as computer controlled machine tools, X-ray diffraction machines, and other industrial and lab equipment. Many of them were sold under the EduSystem marketing program to public schools and universities, and others were used to control laboratory instrumentation. After about 1976, Reuters bought as many as 10,000 OMNIBUS based machines per year, with perhaps 2000 per year going to other customers.

Through the 1980's and 1990's, PDP-8 hardware was frequently discarded or sold for scrap, and many collectors were able to obtain CPU's, peripherals and occasional complete systems in exchange for the effort required to haul them away. This may be changing! In early 1999, a PDP-8 system in unknown condition was sold at auction through eBay for \$1526.00; this is far less than the new cost of such a machine, but far more than the scrap value of such a system! Owners of original DEC hardware will likely need maintenance and test supplies. Douglas Electronics still makes extender boards and breadboards in DEC format, see:

<http://www.douglas.com/hardware/pcbs/breadboards/digital.html>

As of 2000, there are still a modest number of PDP-8 systems in production use, mostly PDP-8/A systems. These are supported by a shrinking number of commercial suppliers and maintenance contractors. For example, Michael Coffey <mdynac@netscape.net> has advertised the availability of spare parts and maintenance documents for Omnibus systems/

By the year 2000, a number of PDP-8 parts and systems have changed hands on E-bay. Sadly, many systems are sold piecemeal, with parts such as

PDP-8 Frequently Asked Questions

By Douglas Jones

core memory destined to hang on den and office walls. The market is spotty, but looking back over the ebay.com sales history is a good way to get an idea of what parts might be worth.

If you can't get real hardware, you can get emulators. Over the years, many PDP-8 emulators have been written; the best of these are indistinguishable from the real machine from a software perspective, and on a modern high-speed RISC platform, these frequently outperform the hardware they are emulating. An emulator is available from DECUS, catalog number RB0128; This and other emulators are available from:

```
ftp://metalab.unc.edu/pub/academic/computer-science/history/pdp-8/emulators
ftp://ftp.cs.uiowa.edu/public/jones/pdp8/emulator.txt.Z
http://www.tiac.net/users/mps/retro/sources/simh25a.zip
(previously ftp://ftp.digital.com/pub/DEC/sim/sources/sim_2.2d.tar.Z)
```

The final collection of emulators listed above includes emulators for the Nova and other DEC machines as well as the IBM 1401.

Bernhard Baehr's emulator for the Apple Mac, complete with an emulated front panel and a fair amount of software is available from:

```
http://home.t-online.de/~bernhard.baehr/pdp8e/pdp8e.html
```

The Spare Time Gizmos emulator for Windows doesn't have the elaborate front panel interface, but it appears to be reasonably complete and has a very realistic teletype interface window.

```
http://www.sparetimegizmos.com/Software/WinEight.htm
```

Finally, you can always build your own. The textbook "The Art of Digital Design," second edition, by Franklin Prosser and David Winkel (Prentice-Hall, 1987, ISBN 0-13-046780-4) uses the design of a PDP-8 as a running example; development new material based on this book continues, including an asynch interface chip and, now, several implementations based on Xilinx FPGAs. Contact Ingo Cyliax or Caleb Hess (cyliax@cs.indiana.edu or hess@cs.indiana.edu) for information on the current state of this work. Other FPGA implementation are being developed by Jon Andrews and David Conroy;
see:

```
http://www.jaywalk.co.uk/pdp8
http://www.jetnet.ab.ca/users/bfranchuk/index.html
```

"Modern VLSI Design - A system approach" by Wayne Wolf (1994 Prentice-Hall)

PDP-8 Frequently Asked Questions

By Douglas Jones

also uses the PDP-8 as a data-path example, as does "Verilog Design Computer Design" by Mark Gordon Arnold (Prentice Hall).

It is worth noting that there are a sufficient number of PDP-8 systems still operational that some companies still manufacture peripherals. For example, Storage Computer (<http://www.storage.com/>) makes RK05 compatible semiconductor "disk drives" that can be directly connected to the Omnibus RK05 controllers of the PDP-8/E,F,M and A.

Occasionally, someone connects a PDP-8 to the internet. The most interesting current venture in that direction is available at:

<http://www.pdp8.net/>

This machine, when working, has complete remote control of the front panel and even the on-off switch from a Java interface, and there's a web-cam so you can see the real machine as it responds to the remote front panel operations.

Subject: Where can I get PDP-8 documentation?

The key documents published by DEC describing each model of the PDP-8 are all out of print, and DEC was in the habit of printing much of their documentation on newsprint with paperback bindings, which is to say, surviving copies tend to be yellow and brittle. DEC distributed huge numbers of catalogs and programming handbooks in this inexpensive paperback format, and these circulate widely on the second-hand market. When research laboratories and electronics shops are being cleaned out, it is still common to find a few dusty, yellowed copies of these books being thrown out.

Douglas Jones has made a small number of bound photocopies of DEC's 1973 introduction to programming, perhaps the definitive introduction to the PDP-8, and the other early DEC handbooks need similar treatment before they all crumble.

Thanks to David Gesswein, a growing collection of PDP-8 documentation, including the Small Computer Handbook, the PDP-8/e/f/m maintenance manual, and prints of various boards have been scanned in and made available on the web at:

<http://highgate.comm.sfu.ca/pdp8/>
http://www.pdp8.net/query_docs/query.shtml

PDP-8 Frequently Asked Questions

By Douglas Jones

Some PDP-8 reference material has been transcribed into Hypertext format and is available over WWW from:

<http://www.cs.uiowa.edu/~jones/pdp8/index.html>

Much more material is available from:

<http://www.spiritone.com/~nabil/pdp8/>

In general, maintenance manuals are hard to find, but valuable. If you need one, you usually need to find someone willing to photocopy one of the few surviving copies. DEC has been friendly to collectors, granting fairly broad letters of permission to reprint obsolete documentation, and the network makes it fairly easy to find someone who has the documentation you need and can get copies. The most difficult to copy material is the large prints, many of which would be quite useful if photoreduced, but this is expensive.

Subject: What operating systems were written for the PDP-8?

A punched paper-tape library of stand-alone programs was commonly used with the smallest (diskless and tapeless) configurations from the beginning up through the mid 1970's. This included a paper-tape based text editor, the PAL III and MACRO-8 assembler, and a FORTRAN compiler, as well as a library of support routines. Many paper tapes from this library survive to the present at various sites! The minimum configuration expected by these tapes is a CPU with 4K memory and a teletype ASR 33 with paper tape reader and punch. Note that much of this paper-tape-based software is based on memory-use and I/O conventions that are incompatible with later disk-based systems.

The DECTape Library System was a DECTape oriented save and restore system that was available from the start. The resident portion of this system occupies only 17 words of memory (7600-7625 octal), and it allowed saving and restoring absolute core images as named files on a reel of DECTape. Initially, program development was still done with paper tape, and only executable memory images were stored on DECTape, but eventually, a limited DECTape-based text editor was introduced, along with a DECTape based assembler.

The 4K Disk Monitor System provided slightly better facilities. This supported on-line program development and it worked with any device that supported 129 word blocks (DECTape, the DF32 disk, or the RF08 disk). It was quite slow, but it also used very little of the available memory.

PDP-8 Frequently Asked Questions

By Douglas Jones

MS/8 or the RL Monitor System, was developed starting in 1966 by Richard F. Lary; it was submitted to DECUS in 1970. This was a disk oriented system, faster than the above, with tricks to make it run quickly on DECTape based systems.

POLY BASIC was a BASIC only system submitted to DECUS and later sold by DEC as part of its EduSystem marketing program. EduSystem 25 Basic is available from:

`ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Edu25Basic/Ascii/`

P?S/8 was developed starting in 1971 from an MS/8 foundation. It runs on minimal PDP-8 configurations, supports somewhat device independant I/O and requires a random-access device for the file system (DECTape is random-access!). P?S/8 runs compatibly on most PDP-8 machines including DECmates, excepting only the PDP-8/S and PDP-5. P?S/8 is still being developed!

Richard F. Lary developed a system called the Fully Upward Compatible Keyboard Monitor; and between a Wednesday and the following Friday, a prototype was up and running from DECTape. The original intention of this project was to build a programming environment for the PDP-8 that looked like TOPS-10 on the PDP-10. A year later, this was released as Programming System/8 (or PS/8), and then renamed OS/8 in 1971 because Eli Glaser (a salesman from Long Island) said he could sell more systems with an operating system than with a programming system, and because, by renaming the system, DEC could increase the price despite Nixon's wage-price freeze.

OS/8, developed in parallel with P?S/8, became the main PDP-8 programming environment sold by DEC. The minimum configuration required was 8K words and a random-access device to hold the system. For some devices, OS/8 requires 12K. There are a large number of OS/8 versions that are not quite portable across various subsets of the PDP-8 family. RX01 images of OS/8 Version 3Q are available, with DEC's free non-commercial use licence, from:

`ftp://ftp.cs.uiowa.edu/public/jones/pdp8/os8distr.txt.Z`
`ftp://ftp.digital.com/pub/DEC/sim/software/os8swre.tar.Z`

The second site above also includes an incomplete but useful RK05 image of OS/8 Version 3R. Parts of the OS/8 source can be found in:

`ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/OS8/`

OS/8 V3D was renamed OS/78 (to match the VT78), and in followups to this distribution, support for Omnibus machines was no longer important. OS/78

PDP-8 Frequently Asked Questions

By Douglas Jones

V4 was developed for the DECmate I, and the name OS/278 used for the versions released with later DECmate machines. These have unnecessary incompatibilities with earlier versions of OS/8. OS/278 and related material is available from DECUS as catalog item 800941, or from:

`ftp://ftp.update.uu.se/pub/pdp8`

`ftp://metalab.unc.edu/pub/academic/computer-science/history/pdp-8/os8`

A growing collection of OS/8 documentation, including the OS/8 software support manual on the internals of the system is available on line from:

`ftp://ftp.dbit.com/pub/pdp8/`

OS8 (no slash) may still be viable. It requires 8K of main memory, an extended arithmetic unit, and DECTape hardware. Unlike most PDP-8 operating systems, it uses a directory structure on DECTape compatible with that used on the PDP-10.

The timesharing system TSS-8 was developed by Don Witcraft and John Everett at DEC, starting in late 1967, and with the first beta sites up and running in the fall of 1968. This was based on a protection architecture proposed by

Adrian Van Der Goor, a grad student of Gordon Bell's at Carnegie-Mellon. It requires a minimum of 12K words of memory and a swapping device; on a 24K word machine, it could give good support for 17 users. It was the standard operating system on the EduSystem 50 which was sold to many small colleges and large public school systems. The first installation was at Lexington High School in Massachusetts, and the second was at Northern Arizona University. Each user gets a virtual 4K PDP-8; many of the utilities users ran on these virtual machines were only slightly modified versions of utilities from the Disk Monitor System or paper-tape environments. Internally, TSS8 consists of RMON, the resident monitor, DMON, the disk monitor (file system), and KMON, the keyboard monitor (command shell).

BASIC was well supported, while restricted (4K) versions of FORTRAN D and Algol were available.

Significant parts of TSS-8 have been found, but at this time, nobody has managed to recover a complete working system. Much of the available TSS/8 code can be found at:

`http://www.spies.com/aek/12bit`

Jim Dempsey, an alum of the OS/8 group at DEC, developed ETOS for Educomp (later Quodata) for the PDP-8/E; this was a true virtual machine operating

PDP-8 Frequently Asked Questions

By Douglas Jones

system in the spirit of IBM's VM/370, and a special board was required to optionally trap JMP and JMS instructions; this was enabled after an emulated CIF instruction so that the actual change of instruction field could be emulated when the JMP or JMS was attempted. After leaving Quodata and founding Network-Systems Design in 1976, Dempsey went on to develop OMNI-8, first installed at Ripon College; initially it was priced at \$4900, several hundred copies were sold. The OMNI-8 operating system supported the enlarged PDP-8 address space of the CESI (Computer Extension Systems Inc) memory cards, and when CESI began making PDP-8 clones, OMNI-8 was extended to support asymmetric multiprocessors (one CPU handled the I/O). The end of OMNI-8 development came around 1990. Dumps of the ETOS kernel and drivers survive in various places, and Jim Dempsey still has the full source for OMNI-8.

Other timesharing systems developed for the PDP-8 include MULTI-8 and MULTOS. The source for MULTOS is available from:

```
ftp://ftp.update.uu.se/pub/pdp8/multos8
```

CAPS-8 was a cassette based operating system supporting PAL and BASIC. There are OS/8 utilities to manipulate CAPS-8 cassettes, and the file format on cassette is compatible with a PDP-11 based system called CAPS-11.

RTS/8 was a real-time system developed by DEC, developed from an earlier system, SRT8, dating back to at least 1974. Curiously, even the last versions of RTS/8 continued to support paper-tape and DECTape. RTS/8 also offered a virtual PDP-8 for background processing, unlike ETOS, this did not require special hardware; instead, software emulation was used to retain control of the machine between the CIF instruction and a following JMP or JMS. Source code for most of the versions of RTS and SRT is available from:

```
ftp://metalab.unc.edu/pub/academic/computer-science/history/pdp-8/rts8
```

WPS was DEC's word processing system, developed for the 8/E with a VT50 terminal with special WPS keycaps replacing the standard keycaps, and widely used on the 1980's vintage machines. It was heavily promoted on the VT-78, and when the DECmates came out, DEC began to suppress knowledge that DECmates could run anything else. WPS-11 was a curious distributed system using a PDP-11 as a file server for a cluster of VT-78 WPS systems. DECmate/WPS Version 2.3 is available from DECUS for the DECmate II and DECmate III under the catalog entry DM0114.

COS-310, DEC's commercial operating system for the PDP-8, supported the

PDP-8 Frequently Asked Questions

By Douglas Jones

DIBOL language. COS-310 was a derivative of MS/8 and OS/8, but with a new text file format. The file system is almost the same as OS/8, but dates are recorded differently, and a few applications can even run under both COS and OS/8. COS was the last operating system other than WPS promoted by DEC for the DECmates.

SCPSYS, developed by D. C. Amoss prior to 1974 at Clemson University, is a system that copies most of the features of LAP (the LINC Assembly Program) for a pure PDP-8 based system. A DECTape of this system has recently come to light, with one known application, Spacewar.

AMOS, an operating system for the PDP-8/E with TD8E DECTape interface, was a very small system developed in Australia or New Zealand and supporting assembly and text editing on a 4K machine.

Subject: What programming languages are supported on the PDP-8

The PAL family of assembly languages, particularly PAL III and PAL8 are as close to a standard assembly language as can be found for the PDP-8; these are included with all OS/8 distributions. They produce absolute object code and there are versions of PAL for minimally configured machines, although these have severe symbol table limitations. Cross assemblers that are somewhat compatible with PAL can be obtained from:

```
ftp://ftp.cs.uiowa.edu/public/jones/pdp8/pal.c.Z
ftp://ftp.cs.uiowa.edu/public/jones/pdp8/palbart.c.Z
ftp://metalab.unc.edu/pub/academic/computer-science/history/pdp-8/emulators/gray
```

MACRO-8 was DEC's first macro assembly language for the PDP-8, but it was rarely used outside the paper-tape environment. MACREL and SABR are assembly languages that produce relocatable output. SABR is the final pass for the ALICS II FORTRAN compiler (developed by ICS); it is included with the standard OS/8 software distributions. Source for these is available from:

```
ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/PS8/FortranSabr/Ascii/
```

MACREL was developed in (unfulfilled) anticipation of similar use. MACREL was heavily used by the DECmate group at DEC. MACREL is available from:

```
ftp://ftp.update.uu.se/pub/pdp8/os8
```

RALF, the relocatable assembler supporting RTPS FORTRAN is also included

PDP-8 Frequently Asked Questions

By Douglas Jones

in OS/8 standard distributions. FLAP, an absolute assembler, was derived from RALF. Both SABR and RALF/FALP are assemblers that handle their intended applications but have quirky and incompatible syntax.

A subset of FORTRAN was supported on both the PDP-5 and the original PDP-8. Surviving documentation describes a DEC compiler written in 1964 by Larry Portner, nicknamed "Fivetrans", and a compiler written by Information Control Systems from 1968. The latter, ALICS II FORTRAN, was originally a paper tape based compiler, but it forms the basis of the OS/8 8K FORTRAN compiler, and was also adapted to the Disk Monitor System (the latter version had overlay support that was never carried forward into more modern systems).

RTPS FORTRAN required 8K and a floating point processor; it had real-time extensions and was a full implementation of FORTRAN IV (also known as ANSI FORTRAN 66). OS/8 F4 is RTPS FORTRAN stripped of the requirement for hardware floating point (if the hardware is missing, it uses software emulation). Versions of FORTRAN is available from

`ftp://ftp.update.uu.se/pub/pdp8/os8`

FOCAL, an interpretive language comparable to BASIC, was available on all models of the family, including the PDP-5 and PDP-8/S. Versions of FOCAL run under OS/8, P?S/8 and other systems, and there were many special purpose overlays for FOCAL developed by DEC and by various users. DEC's later FOCAL releases for the PDP-8 included code to deliberately introduce subtle bugs when run on a DCC 112 computer! Various versions of FOCAL are available from:

`ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Focal/`
`ftp://ftp.cs.uiowa.edu/public/jones/pdp8/coreimages.txt.Z`

Many versions of BASIC were also available, from DEC and other sources. DEC BASIC was widely used on PDP-8 systems sold under the EduSystem marketing program. A paper-tape version was available that ran in 4K and was compatible with disk based systems, versions distributed with OS/8 and TSS-8, an 8K stand-alone time-sharing version was available, and there were others. EduSystem 25 Basic is available from:

`ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Edu25Basic/`

The source code for TSS-8 Basic is available from

`http://www.spies.com/aek/12bit`

DIBOL was DEC's attempt at competing with COBOL in the commercial arena. It was originally implemented under MS/8 but most versions were sold to

PDP-8 Frequently Asked Questions

By Douglas Jones

run under the COS operating system.

Algol was available from a fairly early date. One version is available from:

```
ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Algol/Ascii/
```

At least two Pascal compilers were developed for the PDP-8. One was a Pascal-S interpreter, written in assembler, the other was a Pascal-P compiler with a P-code interpreter written in assembler.

A Pascal S interpreter, requiring a 28K PDP-8/E configuration, is available from:

```
ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Pascal/
```

Another OS/8 Pascal system, the source code for which was rescued by Larwrence LeMay, is available from:

```
ftp://ftp.dbit.com/pub/pdp8/
```

A LISP interpreters was written for the PDP-8; the original version ran in 4K (originally written in Germany?); a disassembled and commented version of this was the basis of expanded versions that eventually could utilize up to 16K. One version of LISP is available from:

```
ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/Langs/Lisp/Ascii/
```

POLY SNOBOL was a version of SNOBOL that was somewhere between Griswold's definitions of SNOBOL 3 and SNOBOL 4.

TECO, the text editor, was included in the standard OS/8 distributions and is a general purpose language (the Emacs editor began as a set of TECO macros!). The story of TECO on the PDP-8 is convoluted. Russ Hamm implemented TECO under his OS8 (without a slash) system, and then gave a listing to Don Baccus at the Oregon Museum of Science and Industry (OMSI) who, along with Barry Smith ported it to PS/8. This was the beginning of what became Oregon Software, later famous for OMSI Pascal.

Richard F. Lary and Stan Rabinowitz made OS/8 TECO more compatible with other versions of TECO, and the result of this work is the version distributed by DECUS (catalog number 110450 is the manual). RT-11 TECO for the PDP-11 is a port of this code.

DECUS also lists the PAGE8 language (catalog numbers 800936), the VISTA editor (catalog number 800938), and the ICE text editor (catalog number 800939).

PDP-8 Frequently Asked Questions

By Douglas Jones

Subject: Where can I get PDP-8 software?

DEC is still making computers, but they've largely forgotten about the PDP-8. The main DEC WWW server is

<http://www.digital.com/>

DECUS, the DEC User Society, is still alive and well, and their submission form still lists PAL8 and FOCAL as languages in which they accept submissions! The DECUS library catalog is available on-line at decus.org; www access is through gopher.decus.org or

<http://www.decus.org/>

To quote the README file from the DECUS on-line catalog, "Items from older DECUS Library catalogs are still also available (provided their media can still be read), but machine readable catalog information is not available for these." Direct questions by E-mail to INFORMATION@DECUS.ORG.

Bob Supnik at DEC has rescued OS/8 from oblivion within DEC and has managed to get DEC to grant a non-commercial free-use licence for OS/8 to all who wish to use it. In addition, he has released a demonstration version of OS/8 for his PDP-8 emulator, available with a copy of the free-use licence from:

<ftp://ftp.digital.com/pub/DEC/sim/software/os8swre.tar.Z>

The following anonymous FTP sites also contain publically accessible archives of PDP-8 software and other information:

<ftp://ftp.update.uu.se/pub/pdp8>
<ftp://russ.ucs.indiana.edu/pub/DEC/PDP8/>
<ftp://ftp.cs.uiowa.edu/pub/jones/pdp8>
<ftp://metalab.unc.edu/pub/academic/computer-science/history/pdp-8>

The latter archive also maintains an archive of traffic in alt.sys.pdp8 in the directory ...pdp8/usenet and an archive of traffic in the pdp8-lovers mailing list in .../pdp8/pdp8-lovers.

The archive at Indiana contains source code for many PDP-8 compilers and interpreters, as well as common utilities and games.

PDP-8 Frequently Asked Questions

By Douglas Jones

Subject: Where can I get additional information?

The companion faq on PDP-8 models and options contains a detailed production history of the PDP-8 as well as a high level description of incompatibilities between models. This is available from:

<http://www.faqs.org/faqs/dec-faq/>
<http://www.cs.ruu.nl/wais/html/na-dir/dec-faq/.html>

A modest attempt at an on-line PDP-8 manual is available from:

<http://www.cs.uiowa.edu/~jones/pdp8/man/>

The mailing list PDP8-Lovers@dbit.com reaches a number of PDP-8 owners and users. This list does not accept postings from nonsubscribers; to subscribe, send mail to PDP8-Lovers-request@dbit.com

The pdp8-lovers mailing list has previously been hosted by onelist.com, mc.lcs.mit.edu, ai.mit.edu, zach1.tiac.net, egroups.com and yahoo.com. The archives for these older lists should be available on dbit.com, but currently, the best archive is at <ftp://zach.dyndns.org/pub/>.

Many "archival" books have included fairly complete descriptions of the PDP-8; among them, "Computer Architecture, Readings and Examples" by Gordon Bell and Allen Newell is among the most accurate and complete, although notationally difficult (see Chapter 5). Gordon Bell has put this on the web at:

http://research.microsoft.com/~gbell/Computer_Structures__Readings_and_Examples/

Subject: What use is a PDP-8 today?

What use is a Model T today? Collectors of both come in the same basic classes. First, there are antiquarians who keep an old one in the garage, polished and restored to new condition but hardly ever used. Once a year, they warm it up and use it, just to prove that it still works, but they don't make much practical use of it.

PDP-8 systems maintained by antiquarians are frequently in beautiful shape. Antiquarians worry about dust, chipped paint, and missing switches, and they establish newsgroups and mailing lists to help them locate parts and the advice needed to fix their machines.

PDP-8 Frequently Asked Questions

By Douglas Jones

In the second class are those who find old machines and soup them up, replacing major parts to make a hotrod that only looks like the original from the outside, or keeping the old mechanism and putting it to uses that were never intended. Some PDP-8 owners, for example, have built PDP-8 systems with modern SCSI disk interfaces! There is serious interest in some quarters in constructing an omnibus board that would support an IDE disk of the variety that was mass-produced for the IBM PC/AT.

Last, there are the old folks who still use their old machines for their intended purposes long after any sane economic analysis would recommend such use. If it ain't broke, don't fix it, and if it can be fixed, why bother replacing it? Both Model T Fords and the classic PDP-8 machines are simple enough that end users can maintain and repair them indefinitely. All you need to keep a vintage -8 running are a stock of inexpensive silicon diodes and a stock of 2N3639B or better, 2N3640 transistors.

Unlike most modern personal computers, PDP-8 systems were routinely sold with complete maintenance manuals; these included schematic diagrams, explanations of not only how to use the devices, but how they are built, and suggestions to those considering building their own peripherals. Compared with many so-called "open systems" of today, the PDP-8 was far better documented and far more open.

Preservation of the PDP-8 has proven to be of immense practical value in defending against the rising tide of patents in the area of interactive graphics. For example, when Sanders Associates sued the Odyssey division of Magnavox, the key testimony in this suit was Steve Russell's Spacewar, originally written for the PDP-1 in the fall of 1961. The fact that documented versions of Spacewar and other computer games dating back to the early 1960's could still be run on a surviving LINC-8 apparently played an important part in arriving at an out-of-court settlement that ended, for practical purposes, the Sanders claim to the technology behind all video games. It is far easier to prove that some software technology existed by demonstrating it on original hardware than by waving a dusty listing in front of someone's face!

Finally, the PDP-8 is such a minimal machine that it is an excellent introduction to how computers really work. Over the years, many students have built complete working PDP-8 systems from scratch as lab projects, and the I/O environment on a PDP-8 is simple enough that it is a very appropriate environment for learning operating system programming techniques.

PDP-8 Frequently Asked Questions

By Douglas Jones

Subject: Who's Who?

You can't beat the book Digital at Work (Digital Press, 1992) for short writeups on the people inside DEC who made the PDP-8!

C. Gordon Bell is generally credited with the original design of the PDP-8 (as well as designing the PDP-4, 5 and 6). He was also involved with recommending what became the PDP-11 when that design was competing with the design that probably became the NOVA, and as vice president of research, he oversaw the development of the DEC VAX family.

Alan Kotok worked with Bell in working up the original specifications of the PDP-8.

Edson DeCastro was a key man in the design of the PDP-5 through the PDP-8/L, then founded Data General to build the Nova.

Ben Gurley designed the early DEC machines, starting with the PDP-1. The actual design work on the -8, however, was done by Ed deCastro, who later founded Data General to build the Nova.

Warren K. Smith was manager of applications engineering for Flip-Chip modules between 1969 and 1975. Much of the M-series TTL module family dates from this period.

Saul B. Dinman, product line manager for the module product line from 1966 to 1969 designed the PDP-8/S and built the engineering prototype, largely in his spare time. Most of his time was devoted to the K-series of industrial automation modules. Later, he founded GRI Computer Corp, where he designed the GRI 909 16-bit minicomputer.

Ken Olson ran DEC from the beginning.

Jozsef Lukacs and Janos Bogdany designed the Hungarian TPA1001 implementation of the PDP-8 instruction set, and Laszlo Szonyi and Pal Karadi designed the TPA/i.

Ed Yourdon, who later became well known as a programming methodology guru, helped hack up the PAL III assembler for the -8 from PAL II.

Richard Merrill invented FOCAL and wrote the original (1968) and classic FOCAL-69 interpreters for the PDP-8. He also did early translations of the interpreter to PDP-7/PDP-9 code and perhaps the earliest PDP-11 version. In addition, he wrote the EDIT-8 paper-tape based text editor based on the FOCAL built-in text editor.

PDP-8 Frequently Asked Questions

By Douglas Jones

Richard F. Lary developed the RL Monitor System, and then went on to develop OS/8, with help from Ed Friedman and another programmer named Paul, under the management of Chuck Conley.

Charles Lasner developed P?S/8, and he is widely known as a leader in the movement to preserve these historic machines. He created the alt.sys.pdp8 newsgroup.

George Thissell oversaw the development of OS/8 FORTRAN-IV, with Denny Pavlock as part of the team.

Wesley Clark developed the LINC while working at Lincoln Labs; this was the first 12 bit minicomputer built with DEC parts.

Mary Allen Wilkes Clark developed the early LAP programs for the LINC.

Don Witcraft wrote the TSS-8 scheduler, command decoder and UUO handler, after working on the first swapping monitor for the PDP-10. John Everett wrote the disk handler, file system, TTY handler and 680-I service routine for TSS-8, after working on the Disk Monitor System and PAL-D, the first disk-based version of PAL.

Roger Pyle and John Everett wrote the PDP-8 Disk Monitor System, and John Everett adapted PAL-III to make PAL-D for DMS. Bob Bowering, author of MACRO for the -6 and -10, wrote an expanded version, PAL-X, for TSS-8.

Jimmy Dykes was the program manager for Harris in the contract development of the Harris 6120 microprocessor; he later moved to GE Semiconductor. Robert M. Smith was involved in the DEC side of this joint venture, after having designed a number of OMNIBUS interfaces during the 1970's.

Douglas W. Jones wrote this FAQ, but prior to the summer of 1992, he'd never used a PDP-8. He has also written a report on how to photocopy and archivally bind ailing paperback books such as DEC's handouts, a PAL-like cross assembler in C, and a UNIX-based PDP-8 emulator.

End of PDP-8 Frequently Asked Questions (posted every other month)
