The PDP-1D

27-Dec-2006

In the early days of DEC's history, the boundary between standard and special systems was fairly open, and the concept of architecture unknown. As a struggling startup, DEC was willing to customize its products to the requirements of its customers, and customers were willing to experiment as well. Consequently, the PDP-1 was extended in multiple different ways, by DEC and by customers, particularly to add time-sharing capabilities.

In 1964, DEC wrote up some of these extensions as the PDP-1D (the –1A was the prototype, the –1B wasn't built, and the –1C was the main production model), but the PDP-1D was more a menu of possibilities than a fixed machine. Two were built, serial #45 for BBN, and serial #48 for Stanford, but they differ in important ways.

The PDP-1D tried to address certain key problems in the PDP-1 architecture:

- Missing operates and skips. Both of the augmented instructions had unused bits, which seemed like a waste.
- Cumbersome character handling. Packing and unpacking characters was a tedious business, involving extensive shifting and rotating of the AC and IO registers.
- 1's complement arithmetic. As floating point and other extended-precision arithmetic representations increased in importance, the 1's complement arithmetic system became a real hindrance. 2's complement was a better representation but lacked hardware support.
- Protection. Time-sharing required basic protection mechanisms in the hardware, to prevent users from damaging the time-sharing executive or each other.
- Clock support. Time-sharing required a real-time clock to time jobs and prevent any single user from monopolizing the machine.
- Multi-terminal support. Time-sharing required attaching multiple terminals to the system.

The PDP-1D required the 16-channel sequence break system, and hardware multiply/divide, as part of its configuration.

Additional Operates and Skips

The PDP-1D used up all the spare bits in both the skip group and the operate group:

```
644000          SNI         skip if IO non-zero

760040          LAI         load (or) AC from IO
```

```
760020          LIA       load (or) IO from AC
760060          SWP       swap AC and IO
770000          CMI       one's complement IO
```

The new skips and operates were present in both serial #45 and serial #48.

Character Handling

The PDP-1, like other 18b machines, used 6b characters, packed three per word. Packing and unpacking characters was a tedious process, involving complex use of AC and IO shifts. The PDP-1D added two new instructions for character handling:

```
012xxxx         LCH       load character
014xxxx         DCH       store character
```

Both used a "byte pointer" consisting of a 2b byte number and a 16b word address:

```
Bits<0:1> = byte number  (0x = 1, 10 = 2, 11 = 3)
Bits<2:17> = word address
```

Because a full 18b word was needed to specify a byte, both LCH and DCH forced "deferred" (indirect) addressing, regardless of whether bit<5> of the instruction was clear or set. Instead, bit<5> was used to signify "automatic" mode, which today we might refer to as auto-increment mode. If the indirect bit was set, the byte pointer was incremented before use by adding 0200000_8. If a carry out occurred, then 1 was added to the address, and the byte number was set to 01. The incremented byte pointer was written back to memory for the next iteration. Thus, in automatic mode, the byte pointer sequenced as follows:

```
00|nnnnnn           start, increment before use
01|nnnnnn           access byte 1
10|nnnnnn           access byte 2
11|nnnnnn           access byte 3
01|nnnnnn+1         access byte 1
10|nnnnnn+2         access byte 2
etc.
```

A peculiar feature of the character handling instructions was "ring mode". Ring mode limited the byte pointer's address increment to the low 3b. This allowed repeated traversal of an 8 word (24 character) ring buffer, presumably a suitable size for low-speed I/O devices like Teletypes.

The additional state for ring mode was kept in a flop that behaved, in some ways, like program flag "zero". To allow for the program flags (and ring mode) to be

saved and restored on a context-shift (essential for time-sharing), a new operate-class, opcode 74, was added:

```
740200          SCI         clear IO
740100          SCF         clear program flags
744000          IIF         or IO from program flags
742000          IFI         or program flags from IO
741000          IDC         index character
```

IDC treated the AC like a byte pointer in automatic mode.

The character handling instructions were present in both serial #45 and serial #48.

2's Complement Arithmetic

The PDP-1's 1's complement arithmetic system reflected common practice of the time, but it proved very cumbersome for multi-precision arithmetic, particularly floating-point. The PDP-1D offered a solution by implementing 2's complement operations in parallel with the standard 1's complement operations.

2's complement arithmetic required retaining the last carry out from an addition. Accordingly, the PDP-1D implemented a Link flag in addition to the Overflow flag. The Link flag behaved like program flag "-1"; it could be saved and restored by IIF and IFI, respectively. 2's complement operations included one memory reference instruction and several new operates:

```
36xxxx          TAD         Link'AC = AC + M[ea] + Link

740020          SZL         skip if Link zero
750020          SNL         skip if Link non-zero
740010          CLL         clear Link
740004          CML         complement Link
740014          STL         set Link
740200          SCM         Link'AC = ~AC + Link
740400          IDA         AC = AC + 1
```

These instructions had a number of peculiarities. First, TAD always added in the Link. This required the Link to be cleared prior to the start of any 2's complement sequence. Second, while SCM could be used as the upper steps of a multi-precision 2's complement, it could not be used as the first step, unless the link was forced to 1, which required a separate instruction (STL occurred after SCM). Finally, IDA, which could be combined with SCM, would not set the Link, and thus could only be used to complement a single word. Further, IDA, like IDC, was subject to Ring Mode, which made it potentially useless for arithmetic.

It took DEC some time to get 2's complement arithmetic right; for example, the PDP-4 could not take a 2's complement with one instruction.

The 2's complement arithmetic capability was present only in serial #45.

Protection

An essential requirement of time-sharing was that users be prevented from accessing or damaging the executive or other users. Rather than provide an "expensive" (in logic terms) base and bounds capability, the PDP-1D implemented a simpler form of protection called "restrict mode." Restrict mode detected the following circumstances:

- Program issues IOT, HLT, or an illegal opcode
- Program accesses a restricted memory bank
- Program issues LCH or DCH in "automatic" mode when the indirect word is 6X7777 (that is, the increment would cross a memory bank boundary)

If a restrict mode violation occurred, the hardware would NOP the current instruction by zeroing the instruction register and force an interrupt to level 16_8.

The PDP-1D didn't really have a concept of monitor mode and user mode. Once restrict mode was turned on, it stayed on, unless any level of the sequence break system was active. Thus, the executive had to run as an interrupt service routine.

Memory protection was on a bank-by-bank basis. If the user program was run with extend mode off, it would be unable to access anything outside its own bank of memory and would operate in a 4KW "virtual machine". If extend mode was on, the user program could still only access specific memory banks, but it would have to know which ones, because there was no relocation mechanism.

Memory protection was implemented differently on serial #45 and serial #48. On serial #45, the granularity of memory protection was 16KW; there were four protection bits, covering the entire physical address space. On serial #48, the granularity was 4KW; there were eight protection bits, covering a maximum of 32KW of memory.

Serial #45 had two additional features. The first was a trap buffer that recorded useful information on a restrict-mode trap. The CPU could read the contents of the trap buffer to help parse the cause of the trap; reading the buffer cleared it. The second was memory renaming. Memory renaming did just that: it changed the upper 2b of the program address to a different value. According to the manual, memory renaming "cannot be bypassed". Thus, it's hard to see how it was used, unless it allowed the non-executive memory banks to be effectively swapped, thereby allowing user programs to run at a fixed base address.

Clock

The clock was a fixed 1Khz 16b counter that was capped at 60,000 (one minute). It generated two interrupts: once a minute, and once every 32ms. The two interrupts were assigned to different interrupt levels. The clock counter could be read, but the clock had no other visible state.

Multi-terminal Support

The terminal multiplexer on the PDP-1D was the Type 630. The Type 630 implemented a scanner over as many as 64 Teletype lines. Each line was half-duplex, with a single buffer and a single ready flag. This made the Type 630 rather hard to program. The line flag could mean either character output complete or character input pending. Software had to track whether the last operation was a send or a receive, and in the case of simultaneous I/O, couldn't really figure out what had happened. (This is a common failing of half-duplex terminal interfaces.)

The PDP-1D documentation on the Type 630 is very sketchy, but fortunately there is a reasonably complete description in the PDP-6 Handbook. The Type 630 was considered workable enough to be adapted for all the early DEC systems, up to and including the PDP-7. For the PDP-8, it was replaced by the Type 680.