Card Readers for DEC's 18b Computers

Punched cards were never a mainstream medium for DEC systems. DEC preferred punched paper-tape, which used less costly peripherals and simpler interfaces. DEC never seemed to be able to get cards quite right. Nowhere is this better illustrated than in the 18b computer line, which implemented seven different card reader options across the five machines in the family (the PDP-1, PDP-4, PDP-7, PDP-9, and PDP-15), as follows:

System	Card Reader	Comments
PDP-1, PDP-4	Type 40	
PDP-7	CR01B	
PDP-9	CR01E	
	CR02B	
PDP-15	CR03B	
	CR15	high speed, data-break
	UC15/CR11	Unichannel PDP-11 card reader

Paraphrasing JBS Haldane, 18b Engineering must have loved card reader controllers, because it made so many of them.

Type 40 Controller

The Type 40 controller for the PDP-1 and PDP-4 set the basic design for programmed I/O card readers. It operated in two modes: Hollerith (alphanumeric) and binary. There was a single interrupt for column data ready; all the other flags (error, card done, end-of-file) were wired to IO status flags.

Instructions (PDP-4):

```
CRSB      706701          skip if column data ready
CRBB      706712          read column data into AC<12:17>
                          and clear column data flag;
                          issued twice in binary mode
CRSA      706704          select (read card) alphanumeric
CRSB      706744          select (read card) binary
```

There were several drawbacks to this interface. First, binary data required two reads and an intermediate shift. Second, there was no interrupt on card done; after reading column 80, software had to poll until the card reader was ready again.

Instructions (PDP-7):

```
CRSB      706701          skip if column data ready
CRBB      706712          read column data into AC<6:17>
```

```
                        and clear column data flag
CRSA      706704        select (read card) alphanumeric
CRSB      706744        select (read card) binary
```

CR01B Controller

The CR01B controller for the PDP-7 used the same instruction set as the Type 40, but fixed one problem: CRRB read 12b of data in binary mode, instead of requiring two reads and a shift. End-of-card detection still required software polling.

CR01E Controller

The CR01E controller for the PDP-9, although nominally a variant on the CR01B, was actually quite different. Most of the status flags were given their own skips. Both column ready and card done caused interrupts. And the select IOT determining the data format, the read IOT determined it.

Instructions (PDP-9):

```
RCSF      706701        skip if column data ready
RCSD      706721        skip if card done
RCSR      706741        skip if reader ready
RCRA      706712        read alpha data into AC<12:17>
                        and clear column data flag
RCRA      706752        read binary data into AC<6:17>
                        and clear column data flag
RCSE      706704        select (read card)
RCLD      706724        clear card done flag
```

This was probably a better programming model than the CR01B, since it allowed alphanumeric versus binary interpretation to be selected at read time; but it was different. Also note, for later reference, that bit <12> was ignored at IOT 4 time; 706744 (CSRB) also worked on the CR01E.

CR02B Controller

The CR02B controller, also for the PDP-9, reverted to the CR01B model of how alphanumeric versus binary selection was done but retained (and extended) the status flag model of the CR01E. End-of-file received its own skip flag.

Instructions (PDP-9):

```
CRSF      706701        skip if column data ready
CRSD      706721        skip if card done
CRSR      706741        skip if reader ready
CRSE      706761        skip if reader end-of-file
```

```
CRBB      706712          read column data into AC<6:17>
                          and clear column data flag
CSRA      706704          select (read card) alphanumeric
CRCD      706724          clear card done flag
CRSB      706744          select (read card) binary
```

The CR02B ignored bit <12> at IOT 2 time; 706752 (RCRA) also worked on the CR02B.

By extraordinary care, PDP-9 software was able to find a common driver that worked for both the CR01E and the CR02B. The common driver depended on two things:

- DEC software never used the alphanumeric mode of operation but instead always translated from 12b binary to internal ASCII.
- The CR01E recognized 706744 as select, in addition to 706704.
- The CR02B recognized 706752 as read column data, in addition to 706712.

Thus, card reads always started with 706744 (read on the CR01E, but read binary on the CR02B), and column data was always read with 706752 (read binary on the CR01E, but read data on the CR02B). Thus both readers always delivered binary data, to a common driver.

CR03B

The fact that alphanumeric mode was never used was not lost on hardware engineering, and the CR03B controller for the PDP-15 took advantage of that to simplify the hardware. The CR03B also shows the impact of using IC's; register bits were now less expensive than flops and skips.

The CR03B implemented a combined status register and data buffer, allowing the driver to sample both at the same time:

Bit	Meaning
0	Reader not ready
1	Hopper empty
2	No pass (no card read – wired to interrupt)
3	Bad data
4-15	Column buffer
16	Card done (wired to interrupt)
17	Column ready (wired to interrupt)

The instruction set was very different from previous controllers:

```
CRSI      706701          skip if reader interrupt
```

```
CROR      706712          read status/data into AC<0:17>
CRSC      706722          clear status/data, select card
CRCS      706704          clear status/data
CRLA      706734          write status/data from AC
```

CR15

All the previous card readers were low-speed, programmed I/O controllers. There must have been some requirement for high-speed card input, because the PDP-15 also offered the CR15, a 3-cycle data-break (or DMA) controller. It used memory locations 22 and 23 as word count and current address respectively.

The CR15 implemented both a status register and a command register. It could operate either as a data-break device or as a programmed I/O device, although the extant drivers use data-break. The status register format was:

Bit	Meaning
4	Photo error
5	Motion error
6	Pick error
7	Hopper empty/stacker full
8	Data missed
9	Trouble (inclusive OR of <4:8>) (wired to interrupt)
10	Busy
11	End-of-file
12	Online
13	Ready
14	Data channel enabled
15	Data channel word count overflow (wired to interrupt)
16	Card done (wired to interrupt)
17	Column ready (wired to interrupt if not data channel)

The command register format was:

Bit	Meaning
13	Clear status register
14	Offset card
15	Interrupt enable
16	Data channel enable
17	Start read

The instruction set was again different from previous units:

```
CRSKP     706701          skip if reader interrupt
CRLD      706712          read column data into AC<6:17>
                          and clear column data flag
```

```
CRLS      706772          read status into AC<0:17>
CRCON     706704          write command register from AC
CRPC      706724          clear status except card done,
                          clear interrupt
```

The driver for XVM/RSX defines different mnemonics, but the meaning is the same.

UC15/CR11

The Unichannel (a PDP-11/05 acting as an I/O processor) of course used the Unibus card reader, the CR11. This need not be discussed in detail, as the interface presented to the PDP-15 was determined by the PIREX software rather than by the underlying hardware.

Summary

While DEC built a card reader interface (or more than one) for every member of the 18b PDP family, it never seemed find a satisfactory programming model for the interface until the very end, when the CR03B dropped Hollerith mode to simplify the hardware, and the CR15 provided data-break for high speed operations. In contrast, the programming model for the paper-tape reader and punch, first established in the PDP-4, remained basically unchanged until the very end.