

PDP-4/X System Reference Manual

dgc@spies.com (David G. Conroy)

Last Updated on Sunday, November 4, 2001 at 4:51:31 PM

1. Introduction

The PDP-4/X is a re-implementation of Digital Equipment Corporation's PDP-4 minicomputer, the archetype for most of the Digital Equipment Corporation's 18-bit systems.

The bulk of the PDP-4/X system is implemented by two FPGAs. The first FPGA (CPU) contains the processor. The second FPGA (IOU) contains the serial I/O interface, a portion of the real-time clock interface, and the IDE disk interface. The memory (which consists of an 32K word memory used by normal programs, and an 32K word memory used only by the front panel program) is implemented by a 64Kx18 static RAM. The PDP-4/X implements an extended memory control modeled after the one in the PDP-7; although both the PDP-4 manual and Bell/Mudge/McNamara talk about an extended memory control for the PDP-4, no description of it seems have survived, and Bell isn't even sure that it was ever designed and/or built. The PDP-4/X also implements an extended arithmetic element modeled after the one in the PDP-7 (mainly to allow the PDP-4/X to run the PDP-9/PDP-15 FORTRAN compiler and object-time system); once again, although both the PDP-4 manual and Bell/Mudge/McNamara talk about an extended arithmetic element for the PDP-4, no description of it seems to have survived.

2. CPU and Memory

The PDP-4 and the PDP-4/X documentation use big endian bit numbering (that is, bit [00] is the most significant bit, and bit [17] is the least significant bit). The PDP-4/X hardware design documents (schematics, FPGA designs) use little endian bit numbering (that is, bit [17] is the most significant bit, and bit [00] is the least significant bit) so that the processor, the static RAM, the EPROM, and the IDE disk can number the bits the same way.

2.1. CPU State

PC[03..17] The PC register holds the address of the next instruction to be executed. Only bits [05..17] of the PC register increment as sequential instructions are executed (the instruction executed after the one at 37777 is at 20000, not at 40000).

AC[00..17] The AC register is the primary arithmetic and logical register. It can be loaded and stored, and serves both as operand and result for arithmetic and logical operations. The AC register can be cleared, complemented, tested, and rotated left or right along with the L flag. The AC register can also be loaded with the contents of the MQ and/or SC registers.

MQ[00..17] The MQ register, part of the extended arithmetic element, holds the multiplier and receives the low half of the product during multiply class instructions, holds the low half of the dividend and receives the quotient during divide class instructions, and serves as an extension to the AC register during shift class instructions. The MQ register can be cleared, complemented, and loaded with the contents of the AC register.

SC[12..17] The SC register, part of the extended arithmetic element, holds the step count for multiply, divide, and shift class instructions. It is loaded with an appropriate (negative) constant at the start of an extended arithmetic instruction, and counts up to 0 during the execution of the instruction.

L The L ("link") flag is a logical extension of the AC register. It is set to 1 if a ones-complement add operation overflows, and it is complemented if twos-complement add operation involving the AC register results in a carry. The L flag can also be cleared, complemented, tested, and rotated left or right with the AC register.

IE The IE ("interrupt enable") flag is 1 if the processor is running with interrupts enabled, and is 0 if it is running with interrupts disabled.

EM[00..01] The EM ("extend mode") register is 00 if the processor is running in normal indirect addressing mode, 10 if the processor is running in extended indirect addressing mode, and 11 if the processor is running in extended indirect addressing mode and armed to do a restore. The EM register never contains 01.

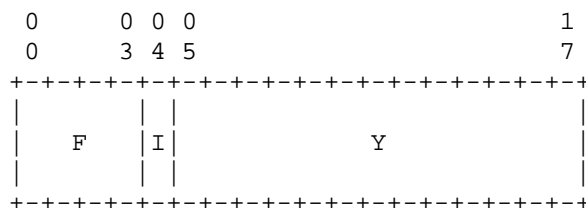
SW[00..17] The SW ("switch") register emulates the switches normally on the front panel.

IM The IM ("instruction mode") flag is 0 when the processor is architecturally halted, and is 1 when the processor is architecturally running. When the processor is halted the front panel program, located in front panel memory, is in control. When the processor is running a normal program, located in normal memory, is in control.

DM The DM ("data mode") flag, if set to 1, forces the processor to take indirectly addressed memory operands from normal memory without regard for the state of the IM flag. It is set to 1 by the front panel program when it is necessary to access normal memory.

HM The HM ("halt mode") flag, if set to 1, forces the processor to halt. It is set to 1 by the front panel program to implement single stepping.

2.2. Memory Reference Instructions



All memory reference instructions compute their effective address in the same way. If the I field is 0 then the instruction is said to be directly addressed; the Y field specifies the effective address, which is in the same 8K field as the instruction. If the I field is 1 then the instruction is said to be indirectly addressed; the Y field specifies the location of an indirect word, in the same 8K field as the instruction, the interpretation of which is a function of bit [00] of the EM register. If bit [00] of the EM register is 0 then bits [05..17] of the indirect word specify the effective address, which is in the same 8K field as the instruction. If bit [00] of the EM register is 1 then bits [03..17] of the indirect word specify the effective address, which is anywhere in memory.

Any indirect word fetched from a location with an offset between 00010 and 00017 in any 8K field is incremented before it is used. In addition, if an instruction specifies a location with an offset between 00010 and 00017 then the location is forced to be in field 0 (if an indirect word specifies a location with an offset between 00010 and 00017 things work normally). Note that this is not how the PDP-4 and PDP-7 worked (they had auto-incrementing locations in every field), but it is how the PDP-9 and PDP-15 worked, and there is code in the PDP-9/PDP-15 operating system which cares, so the PDP-4/X was changed.

- | | | |
|------|-----|---|
| 0001 | DAC | The AC register is stored into the memory location specified by the effective address. |
| 0010 | JMS | The L flag, bit [00] of the EM register, and the PC register (which has been incremented to point to the instruction after the JMS) is stored into the memory location specified by the effective address (L in bit [00], EM[00] in bit [01], constant 0 in bit [02], and PC in bits [03..17]). The effective address, with bits [05..17] incremented, is then loaded into the PC register. |
| 0011 | DZM | The constant 0 is stored into the memory location specified by the effective address. |
| 0100 | LAC | The memory location specified by the effective address is loaded into the AC register. |
| 0101 | XOR | The memory location specified by the effective address is combined with the AC register using a bit-by-bit XOR. |
| 0110 | ADD | The memory location specified by the effective address is combined with the AC register using ones-complement addition (that is, with an end-around carry). If the addition overflows (that is, if after the end-around carry has been propagated, the sum of two like-signed numbers has the opposite sign) the L flag is set (otherwise it is not |

changed). A result of negative 0 (777777) is not treated as a special case; it happens when -0 is added to -0, -0 is added to +0, and -N is added to +N.

- 0111 TAD The memory location specified by the effective address is combined with the AC register using ones-complement addition. If the addition results in a carry out of bit [00] the L flag is complemented.

- 1000 XCT The memory location specified by the effective address is executed. The PC is not effected by XCT, so if the target of an XCT is a skip then the instruction after the XCT instruction is skipped, and if the target of an XCT is a JMS the address saved is that of the instruction after the XCT instruction.

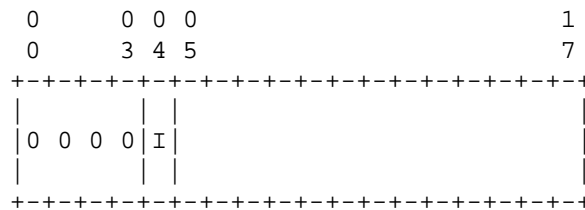
- 1001 ISZ The memory location specified by the effective address is incremented, and if the result is 0, bits [05..17] of the PC register are incremented, skipping the next instruction.

- 1010 AND The memory location specified by the effective address is combined with the AC register using a bit-by-bit AND.

- 1011 SAD The memory location specified by the effective address is compared with the AC register, and if they are not equal, bits [05..17] of the PC register are incremented, skipping the next instruction.

- 1100 JMP The effective address is loaded into the PC register. If the IM flag is set to 1 a JMP instruction which specifies indirect addressing also performs a restore if EM[01] is set to 1; bit [00] of the indirect word is loaded into the L flag, bit [01] of the indirect word is loaded into EM[00], and EM[01] is set to 0.

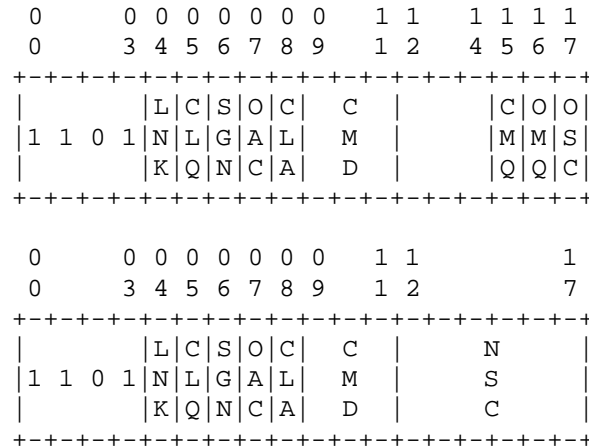
2.3. CAL Instruction



The CAL instruction performs the same function as the JMS instruction, except that it computes the direct portion of the effective address in a different way which allows bits [05..17] of the instruction to be used by software.

If bit [00] of the EM register is 0 then the direct portion of the effective address is location 00020 in the same 8K field as the instruction. If bit [00] of the EM register is 1 then the direct portion of the effective address is location 00020 in field 0. Indirect addressing works normally.

2.4. EAE Instruction



The EAE instruction is used control the extended arithmetic element. There are two sub-formats of the EAE instruction, one used when the CMD field is 000 (SETUP), and the other used when the CMD field is not 000 (not SETUP).

The extended arithmetic element is quite strange, and although this section describes how the PDP-4/X interprets the EAE instruction, it does not describe the stylized code sequences and combinations of micro-operations which implement the various signed and unsigned operations. These should be added to this document, but for now the appropriate sections of the manuals for the 18-bit processors (the manuals for the PDP-9 and PDP-15 are available at a number of sites on the web) will need to be consulted for enlightenment.

All EAE instructions begin with a setup phase, controlled by bits [04..08], and, if the CMD field is 000 (SETUP), bits [15..17], of the instruction word. Each bit specifies a micro-operation, executed in the order LNK/CLQ/SGN, then OAC/CLA, then (possibly) CMQ/OMQ/OSC.

- [04] LNK The L flag is loaded with the contents of bit [00] (that is, the sign bit) of the AC register.
- [05] CLQ The MQ register is cleared.
- [06] SGN The (temporary) EAE sign1 flag is loaded with the contents of bit [00] (that is, the sign bit) of the AC register. Then, if the EAE sign1 flag is set to 1 and the OAC bit is set to 0, the AC register is (ones) complemented.
- [07] OAC The MQ register is loaded with the inclusive-or of the MQ register and the AC register.
- [08] CLA The AC register is cleared.
- [15] CMQ The MQ register is (ones) complemented.
- [16] OMQ The AC register is loaded with the inclusive-or of the AC register and the MQ register.

[17] OSC The AC register is loaded with the inclusive-or of the AC register and the SC register (zero extended to 18 bits).

If the CMD field is not 000 (SETUP) then the EAE instruction proceeds into the execution phase. The execution phase begins with some additional setup. First the SC register is loaded with the (twos) complement of the NSC field of the instruction. Next, the (temporary) EAE sign2 flag is loaded with the contents of the L flag. Next, if the CMD field is 001 (MUL) or 011 (DIV), and the EAE sign1 flag is set to 1, the MQ register is (ones) complemented. Finally, if the CMD field is 001 (MUL) or 011 (DIV) an operand word, located immediately after the EAE instruction word, is fetched, and the PC register is incremented.

The PDP-4/X actually executes this additional execution phase setup in parallel with the setup phase.

The execution phase consists of a variable number of steps, with the function of each step controlled by the CMD field. At the end of each step the SC register is incremented, and another step begins if the SC register is non-zero. Because the SC register is incremented and tested at the end of each step the execution phase of an instruction with an NSC field of 0 is 64 steps long.

001 MUL If bit [17] of the MQ register is set to 1 then the operand is added to the AC register. Next the AC and MQ registers are shifted right. Bit [00] of the AC register is loaded with any carry generated by the (conditional) add operation. The bit shifted out of bit [17] of the MQ register is lost. The L flag is set to 0.

011 DIV If the contents of the AC register is greater than or equal to the operand then the operand is subtracted from the AC register and the (temporary) quotient bit is set to 1; otherwise the (temporary) quotient bit is set to 0. If this is the first execution step and the quotient bit is set to 1 (divide overflow) then the L flag is set to 1 and the execution phase ends; the AC and MQ registers are unspecified. Otherwise the AC and MQ registers are shifted left (unless this is the last execution step, when only the MQ register is shifted left). Bit [17] of the MQ register is loaded with the quotient bit. The L flag is set to 0.

100 NORM The AC and MQ registers are shifted left. Bit [17] of the MQ register is loaded with the contents of the L flag. The bit shifted out of bit [00] of the AC register is lost. The NORM instruction differs from the LLS instruction in that an execution step does not begin if bit [00] and bit [01] of the AC register are different (the NORM instruction is unique in that its execution phase can have no steps, which happens when a NORM instruction is given and bit [00] and bit [01] of the AC register are different).

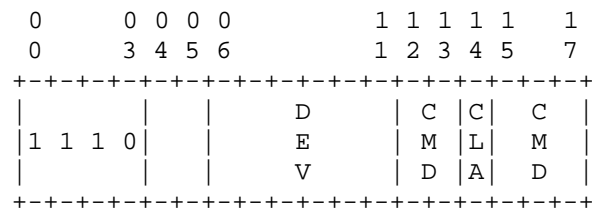
101 LRS The AC and MQ registers are shifted right. Bit [00] of the AC register is loaded with the contents of the L flag. The bit shifted out of bit [17] of the MQ register is lost.

- 110 LLS The AC and MQ registers are shifted left. Bit [17] of the MQ register is loaded with the contents of the L flag. The bit shifted out of bit [00] of the AC register is lost.
- 111 ALS The AC register is shifted left. Bit [17] of the AC register is loaded with the contents of the L flag. The bit shifted out of bit [00] of the AC register is lost.

The cleanup phase begins after the execution phase. If the CMD field is 001 (MUL) and the EAE sign1 and EAE sign2 flags are different, then the AC register and the MQ register are (ones) complemented (the product gets the appropriate sign). If the CMD field is 011 (DIV) and the EAE sign1 and EAE sign2 flags are different, then the MQ register is (ones) complemented (the quotient gets the appropriate sign). If the CMD field is 001 and the EAE sign1 flag is set to 1 then the AC register is (ones) complemented (the remainder always has the same sign as the dividend).

The PDP-4/X actually executes the cleanup phase in parallel with the final step of the execution phase.

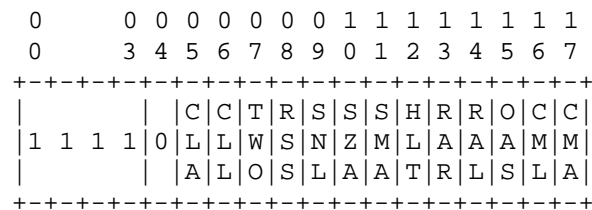
2.5. IOT Instruction



The IOT instruction is used to transfer data in an out of the processor via the AC register. The DEV field specifies the device. The CMD fields specifies the (device dependent) command. The CLA bit specifies that the AC register should be cleared before the IOT command is executed. Bits [04..05] are ignored, although they should be set to 00 (IOT instructions with bits [04..05] not set to 00 were eventually used for instruction set extensions on the PDP-15).

IOT instructions which specify nonexistent devices are no-operations, except for any effect on the AC register caused by the CLA bit.

2.6. OPR Instruction



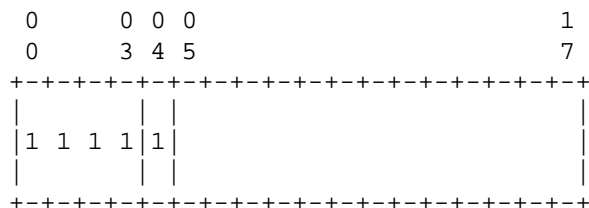
Each bit in the OPR instruction specifies a micro-operation, executed in the order SNL/SZA/SMA, then CLA/CLL, then CMA/CML, then OAS, then RAL/RAR/RTL/RTR, then HLT. On a real PDP-4 the RAL/RAR micro-operations could not be combined with CMA/CML/OAS, and the RTL/RTR micro-operations could not be combined with CLA/CLL/CMA/CML/OAS. The PDP-4/X has none of these restrictions.

- [05] CLA The AC register is cleared.

[06]	CLL	The L flag is cleared.
[07]	TWO	If a rotate is specified by bit [13] or bit [14], then that rotate is by two bits rather than by one bit.
[08]	RSS	If this bit is 0 then the processor skips if any of the skip conditions are true. If this bit is 1 then the processor skips if none of the skip conditions are true.
[09]	SNL	The skip condition is true if the L flag is 1.
[10]	SZA	The skip condition is true if the AC register is 0.
[11]	SMA	The skip condition is true if the AC register is less than zero (that is, if AC[00] is 1).
[12]	HLT	A halt interrupt is generated. Control passes to the front panel program.
[13]	RAR	The AC register and L flag are rotated right by one or two bits, depending on the value of the TWO bit.
[14]	RAL	The AC register and L flag are rotated left by one or two bits, depending on the value of the TWO bit.
[15]	OAS	The AC register is replaced with the bit-by-bit inclusive-or of the AC register and the SW register.
[16]	CML	The L flag is complemented.
[17]	CMA	The AC register is (ones) complemented.

The result of an OPR instruction which specifies both an RAR and an RAL micro-operation, or which specifies a TWO micro-operation without specifying an RAR or an RAL micro-operation, is undefined.

2.7. LAW Instruction



The LAW instruction is used to load constants. The entire instruction word, including bits [00..03] (the operation code), are loaded into the AC register.

2.8. Operating Modes

The PDP-4/X does not implement a "switches and lights" front panel. Instead, it implements a front panel program which performs all of the functions normally provided by the front panel, driven by commands from the console terminal. Implementing the front panel program is tricky because there is no place to hide it in the PDP-4/X address space, and because the EM register (especially bit

[01], the bit set by the EMIR instruction) and the IE flag are tricky to save and restore.

The PDP-4/X operates in one of two modes, controlled by the IM flag. If the IM flag is 0 then the processor is architecturally halted, and the front panel program is being executed. If the IM flag is 1 then the processor is architecturally running, and a normal program is being executed. In both cases the processor is executing normal instructions; instruction execution never really halts.

If the IM flag is 0 the EM register and the IE flag are ignored; the processor behaves as if the EM register contains 10, and the IE flag contains 0. Although instructions which explicitly read and write the EM register or the IM flag read and write the real registers, indirect JMP instructions do not alter the EM register. The idea is that the EM register and the IE flag hold their normal values as the front panel program executes, and the front panel program can read and write them by explicit action.

The PDP-4/X has two 32K memory spaces. One space ("normal space") is used by normal programs. The other space ("front panel space") is used only by the front panel program.

The IM flag determines which of the memory spaces is used for instructions, indirect words, and directly addressed operands. If the IM flag is 0 front panel space is used. If the IM flag is 1 normal space is used.

The DM flag determines which of the memory spaces is used for indirectly addressed operands. If the DM flag is 0 the space specified by the IM flag is used. If the DM flag is 1 then normal space is used. Normally the DM flag is 0, but the SDM instruction sets it to 1 for the duration of the next instruction, which allows the front panel program to (selectively) access normal space.

The SIM instruction sets the IM flag to 1. There is a one instruction delay so that the final JMP is fetched from front panel space.

The SHM instruction sets the IM flag to 1, and sets the HM flag to 1 for the execution of a single instruction, which forces a halt interrupt after a single instruction is executed. There is a one instruction delay so that the final JMP is fetched from front panel space, and so that the halt interrupt happens after the final JMP. This feature is used by the front panel program to perform single stepping.

2.9. Interrupts

A normal interrupt is generated at the end of an instruction if the instruction is not an IOT, if interrupts are enabled (the IE flag is 1, which implies that the processor is architecturally running) and if some device is making an interrupt request. If all of these conditions are true the processor stores the value of L/EM[00]/PC into location 0 of normal space, sets the IE flag to 0, sets the EM register to 00, and sets the PC to 00001.

A halt interrupt is generated at the end of an instruction if the processor is architecturally running and there is a halt request (either the last instruction executed was an OPR with the HLT bit set, or the HM flag is 1, or the external halt request signal is asserted). If both of these conditions are true the

processor stores the value of L/EM[00]/PC into location 0 of front panel space, sets the IM flag to 0, sets the DM flag to 0, and sets the PC to 00001.

If there are multiple interrupt requests at the end of an instruction a halt interrupt has the higher priority than a normal interrupts.

Note that all interrupts happen at the end of instructions, so if something happens which prevents an instruction from reaching the end (an infinite chain of XCT instructions, or an IOT instruction which is externally stalled) the processor is effectively hung (just like the real thing).

2.10. Internal IOT Instructions

The processor uses IOT instructions with device code 00 to control the interrupt system. Any instruction with bit [16] set to 1 sets the IE flag to the value of bit [12]. The following combinations are standard and generally considered useful.

700002 IOF Set the IE flag to 0.

700042 ION Set the IE flag to 1. Note that because ION is an IOT instruction interrupts are blocked between the ION instruction and the next instruction, which is usually a JMP I 0, sending control back to the point of interruption.

The processor uses IOT instructions with device code 00 and bits [15..17] set to 0 (which would generate no IOP pulses, and which could not, therefore, be used to control a real I/O device on a real PDP-4) to control the front panel emulation hardware. Bits [12..13] are used as a function code.

700000 LSA Load the contents of the AC register into the SW register.

700020 SDM Set the DM flag to 1 for execution of a single instruction.

700040 SIM Set the IM flag to 1 after a one instruction delay.

700060 SHM Set the IM flag to 1, and the HM flag to 1, for the execution of a single instruction, after a one instruction delay.

The system contains an interval clock, distributed between the CPU and IOU. If the clock is enabled the IOU generates a 60 Hz signal and sends it to the CPU, which increments location 7 in response to every rising edge. If the result of the increment is 0 (that is, if there is a carry out of the adder) the CPU sends a clock overflow signal back to the IOU, which sets the clock flag. The CPU only allows location 7 to be incremented when the IM flag is set to 1.

The processor uses IOT instructions with device code 00 to control the interval clock. Any instruction with bit [17] set to 1 skips if the clock flag is set, and any instruction with bit [15] set to 1 clears the clock flag and sets the clock enable to the value of bit [12]. The following combinations are standard and generally considered useful.

700001 CSF Skip if the clock flag is set to 1.

700004 COF Set the clock flag to 0, and disable the clock.
700044 CON Set the clock flag to 0, and enable the clock.

The clock rate is fixed at 60 ticks per second, which is generated by dividing down a 1.8432 MHz oscillator.

The processor uses IOT instructions with device code 77 to control the memory extension logic. These instructions are microcoded. Any instruction with bit [17] set to 1 skips if the appropriate bit of the EM register is set to 1. Any instruction with bit [16] set to 1 sets bit [00] of the EM register, and sets bit [01] of the EM register to the value of bit [12]. Any instruction with bit [15] set to 1 sets bits [00..01] of the EM register to 0. The operations controlled by bit [16] happen before the operations controlled by bit [15], so if both bit [15] and bit [16] are set, bit [15] determines the result. The following combinations are standard and generally considered useful.

707701 SEM If the IM flag is set to 0 skip if bit [01] of the EM register is set to 1, which means the processor is armed to perform an interrupt restore. If the IM flag is set to 1 skip if bit [00] of the EM register is set to 1, which means the processor is in extended indirect addressing mode.
707702 EEM Set the EM register to 10, which puts the processor into extended indirect addressing mode.
707742 EEMR Set the EM register to 11, which puts the processor into extended indirect addressing mode, and arms the processor to perform a restore at the next indirect JMP.
707704 LEM Set the EM register to 00, which puts the processor into normal indirect addressing mode.

The EEMR instruction is a bit of a hybrid. The PDP-7 and PDP-9 have EMIR at this code point; EMIR sets EM to 1, and arms the processor to restore EM at the next indirect JMP (actually the restore can only set EM to 0 on the PDP-7, but since EMIR sets EM to 1, it takes a fairly contrived program to notice). The PDP-15 has RES at this code point; RES arms the processor to restore L and EM at the next indirect JMP (remember that a PDP-15 is always in extend mode). Having EEMR arm the processor to restore the L flag (like the PDP-15) makes EEMR a useful substitute for DBR, which is used all over the PDP-9/PDP-15 advanced system software. Note that the PDP-9/PDP-15 advanced system software actually uses EMIR/RES, so it can't care (much) if it restores the L flag.

3. External I/O

The PDP-4/X system includes a teletype port, a disk port, and an expansion port.

3.1. Teletype Port

The teletype port interfaces with devices that comply with the RS-232 specification. The teletype port sends and receives 8-bit characters, even though the serial port on the PDP-4 did not, so that it can be used to communicate with ASCII devices.

The receive side of the teletype port emulates the keyboard on device code 03. Any instruction with bit [17] set to 1 skips if the keyboard flag is set, and any instruction with bit [16] set to 1 clears the keyboard flag and OR's data into the AC register. All combinations are legal. The following are standard and generally considered useful.

700301	KSF	Skip if the keyboard flag is 1.
700312	KRB	Load the contents of the keyboard buffer into the AC register, and set the keyboard flag to 0. The flag will set to 1 when the next character is received.

Although the teletype port is data leads only, the state of the keyboard flag can be read on the RTS pin (the active-low RTS signal is true if the receive flag is false). This allows a terminal emulator, which may be sending a file, to avoid sending characters when the PDP-4/X is not ready to read them (assuming, of course, that it can stop quickly enough).

The transmit side of the teletype port emulates the printer on device code 04. The instruction is microcoded, with bit [17] skipping if the flag is set, bit [16] clearing the flag, and bit [15] sending the character. All combinations are legal. The following are standard and generally considered useful.

700401	TSF	Skip if the printer flag is 1.
700402	TCF	Set the printer flag to 0.
700406	TLS	Begin printing the character whose code is in bits [10..17] of the AC register, and set the printer flag to 0. The flag will set to 1 when the character has been printed.

The serial port data rate is fixed at 9600 bits per second, which is generated by dividing down a 1.8432 MHz oscillator.

At reset the keyboard flag and the printer flag are set to 0. This is not a useful initial state for the printer flag (which really wants to be set to 1), but it is the way a real PDP-4 worked. The front panel program fixes this by printing a null character.

3.2. Disk Port

The disk port interfaces with disks that comply with the IDE specification. The port is very simple, and only supports type-0 programmed I/O timing (the slowest mode).

Only the 9 least significant bits of the disk data bus are connected to the disk port; the most significant 7 bits of the disk data bus are gently pulled down with (separate) resistors. This somewhat unusual hookup, which was done for pin count reasons (the IOU is in an 84 pin package), does not make the disk driver any more complex and/or slower, but does make it impossible to use disk commands demand that all bits of the disk data bus be read or written (for example, identify device) unless the disk supports the (optional) 8-bit data bus feature.

Bit [12] of the disk port IOT instruction determines if the 9 bits of data appear in bits [09..17] (bit [12] = 0) or in bits [00..08] (bit [12] = 1) of the

word read or written by the IOT instruction. On the reads the unused bits are guaranteed to be 0. On writes the unused bits are discarded.

The disk port uses device codes between 40 and 47, which do not seem to conflict with any standard assignment.

704x00	HSI	Skip if the IDE interface is requesting an interrupt; that is, if the INTRQ signal is asserted. Note that the INTRQ signal has a pull-down, so that if the INTRQ driver is floated at the drive (by setting NIEN in the drive control register to 1) INTRQ appears false.
704x14	HRR	Read a register in the IDE command register block (that is, a register in the space which is accessed with CS0 asserted and CS1 negated). Bits [09..11] of the instruction word supply DA[02..00] to the device.
704x05	HWR	Write a register in the IDE command register block (that is, a register in the space which is accessed with CS0 asserted and CS1 negated). Bits [09..11] of the instruction supply DA[02..00] to the device.
704x16	HRS	Read a register in the IDE control block (that is, a register in the space which is accessed with CS0 negated and CS1 asserted). Bits [06..08] of the instruction supply DA[02..00] to the device, but the only sensible value is 6, which accesses the alternate status register.
704x07	HWC	Write a register in the IDE control block (that is, a register in the space which is accessed with CS0 negated and CS1 asserted). Bits [09..11] of the instruction supply DA[02..00] to the device, but the only sensible value is 6, which accesses the device control register.

The intention is that each 18-bit word is stored in the least significant 9 bits of two 16-bit disk words. The sequence for writing an 18-bit word to the disk port is "LAC data; HWR+40; HWR" (the first HWR has bit [12] = 1, so bits [00..08] are sent to the disk; the second HWR has bit [12] = 0, so bits [09..17] are sent to the disk). The sequence for reading an 18-bit word from the disk is "HRR+40; HRR-10; DAC data" (the first HRR has bit [12] = 1 and bit [14] = 1, so it loads bits [00..08] from the disk; the second HRR has bit [12] = 0 and bit [14] = 0, so it or's in bits [09..17] from the disk).

3.3. Expansion Port

The expansion port is a header attached to the same buses as the serial port and the disk port. Software transfers data to and from a device attached to the expansion port using programmed I/O.

Expansion port devices should be designed to clear all interrupt requests when they are reset so the front panel program need not be updated.

3.4. IORS Instruction

The IORS instruction allows software to read the state of multiple I/O device flags, and the state of the IE flag, at once.

700314 IORS Load a summary of the state of the I/O flags into the AC register. This word has the processor IE flag in bit [00], the teletype keyboard flag (tested by KSF) in bit [03], the teletype printer flag (tested by TSF) in bit [04], the clock flag (tested by CSF) in bit [06], and the clock enable flag (manipulated by CON/COF) in bit [07]. All other bits are 0.

The IORS instruction is implemented by the IOU, which supplies all 18 bits of data (and makes it essentially impossible for a device attached to the expansion port to participate). The CPU sends the state of the IE flag to the IOU so that the IOU can insert it into bit [00].

4. Front Panel Program

When the PDP-4/X is reset 8K words are copied from an EPROM into field 0 of the special front panel memory, the IM flag is set to 0, the DM flag is set to 0, the HM flag is set to 0, and the PC is set to 00000. A JMP at location 00000 in front panel space sends control to the cold start entry point of the front panel program. The cold start entry point initializes the CPU, initializes the I/O devices, and begins parsing commands.

When the PDP-4/X halts the PC register is stored in location 0 of the front panel memory, and control passes to location 1.

The front panel program prompts for command lines with ">>>", erases single characters with delete, erases the whole line with control-U, and processes the command line on carriage return. All numbers are octal.

The following commands are implemented in the latest version (PANEL.008) of the front panel program.

- A [d] Examine or deposit the AC register.
- B [a] Breakpoint. If an "a" argument is present then a breakpoint is set at that address. If no argument is present then the addresses of all breakpoints are printed. Only the least significant 15 bits of the "a" argument are used.
- C [a] Clear breakpoint. If an "a" argument is present then the breakpoint at that address is deleted. If no argument is present than all breakpoints are deleted. Only the least significant 15 bits of the "a" argument are used.
- D a [d ...] Deposit memory. Only the least significant 15 bits of the "a" argument are used.
- E a [n] Examine memory. The display is in octal, eight words per line. Only the least significant 15 bits of the "a" argument are used.

F a n d [m] Find words in memory. Memory, starting at location "a" and extending for "n" words is searched for data "d". Only the least significant 15 bits of the "a" argument are used. If an "m" argument is present the data from memory is and'ed with "m" before it is compared with "d".

G [a] Go. If an "a" argument is specified the state of the machine is reset (the PC register is set to the least significant 15 bits of the "a" argument, the IE flag is set to 0, and the EM register is set to 00).

I [d] Examine or deposit the IE flag. Only the least significant bit of the "d" argument is used.

L [d] Examine or deposit the L flag. Only the least significant bit of the "d" argument is used.

M [d] Examine or deposit the EM register. Only the least significant 2 bits of the "d" argument are used.

P [d] Examine or deposit PC register. Only the least significant 15 bits of the "d" argument are used.

Q [d] Examine or deposit the MQ register.

R Load a binary file. The file is in the format invented for the PDP-1/X, which is unlike any Digital format.

S [d] Examine or deposit the SC register. Only the least significant 6 bits of the "d" argument are used.

T Trace. A single instruction executed, after which the processor is forced to halt, which forces control back to the front panel program.

U a [n] Examine memory in octal and as instructions.

V Version. The front panel program version message is printed; this is the same message which is printed when the front panel program starts upon reset.

W [d] Examine or deposit the SW register.

X op [d] Execute the instruction specified by the "op" argument. If a "d" argument is specified then that value is loaded into the AC register before the instruction is executed, otherwise 0 is loaded into the AC register before the instruction is executed. The value of the AC register after the instruction is executed, and an indication if the instruction results in a skip, is printed. The instruction is usually an IOT instruction, although any instruction can be specified. Note that the instruction is executed in front panel mode, so memory reference instructions will access front panel space, which is generally not useful (although it can be used to peek at variables when debugging a front panel program).

Z Load bootstrap. The bootstrap device handler for the disk is copied into memory locations 77637 through 77777 (the locations where the DEctape bootstrap would normally go on a system with 32K words of memory). The Z command also sets the PC register to 77646 and the EM register to 2 so that the operating system can be started by a G command with no arguments.

The front panel program can print the following error messages.

?ILL CMD Illegal command. This message is displayed if a completely unrecognized command is entered.

?SYN ERR Syntax error. This message is displayed if a command is recognized by there is something wrong with the command's arguments.

?BPT ERR Breakpoint error. This message is displayed if an attempt is made to set a breakpoint when the breakpoint table is full, to set more than one breakpoint at the same address, or to clear a breakpoint which does not exist.