PDP-1 COMPUTER
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

PDP-33

INPUT/OUTPUT IN THE PDP-1-X

J. B. Dennis

March 10, 1966

Input/Output in the PDP-1-X

A general and flexible input/output scheme has been devised for the expanded time-shared arrangement for the EE-RLE PDP-1 computer system. The major objectives of this scheme are:

1) To make possible modular additions of input/output equipment without requiring alterations of previously installed modules, e.g. the PDP-1 processor.

2) To satisfy the protection, assignment and scheduling requirements of time-shared operation in a uniform way that allows for orderly expansion while putting as little limitation as possible on the object computations that may be run.

Physically, revised input/output control logic in the PDP-1-X processor links programs with peripheral devices through an input/output bus that links the peripherals with the PDP-1-X processor as shown in Fig. 1. The bus consists of 36 coaxial conductors, that are connected identically to each i/o module. Up to 64 i/o modules may be connected. The physical length of the bus is limited by timing considerations to about 500 feet, and perhaps more severely by the attennation characteristics of the cable.

Logically, each set of input/output operations whose timing must be controlled independently of others is associated with a distinct input/output function, of which provision has been made for 64. Each i/o function may have up to sixteen variants which are associated with the distinct operations belonging to the same function.

Each input/output step by a process is the execution of some variant of an i/o function by the instruction ivk described below. The execution may call for a pause in which case the process is suspended until an i/o function done signal is received from the peripheral controller. (For programming of concurrent i/o, see below.) Otherwise, completion of execution is immediate and operation of the process continues. Execution of one variant of an i/o function must be completed before another variant of the same i/o function may be initiated. On the other hand, any or all i/o functions may be in execution (initiated but not done) simultaneously.

Programming

In non-time-shared operation input/output operations are controlled by iof instructions with the format shown in Fig. 2. Control of timing may be accomplished by in-out wait (suspension of processor operation) or by means of processor interruption, according to the mode bits of the iof instruction.

mode	bit 7	bit 8	
pause	0	0	Processing is suspended until the most recent wait mode i/o function has completed. Execution of the current i/o function variant is started. Processing is suspended (if necessary) until execution of the current i/o function is complete.
break	0	1	Processing is suspended until the most recent wait mode i/o function has completed. Execution of the current i/o function variant is started. Processing is continued whether or not execution of the current i/o function is complete. If not complete a distinguishable interrupt will occur when the function done signal is received from the peripheral.
wait	1	--	Processing is suspended until the most recent wait mode i/o function has completed. Execution of the current i/o function variant is started and processing is continued whether or not execution of this i/o function is complete. No interrupt will occur.

The wait mode of control is provided so that computation may overlap i/o wait for such devices as the CRT display where the i/o delay is too brief to warrant sequence break programming.

A sequence break resulting from i/o function completion will produce a one in bit-? of the check status word. Execution of an rfn instruction -- read function number -- places the six-bit number of the completed i/o function in bits 12-17 of the IO register, and clears the status bit to permit further function done signals to be accepted.

In those cases where an i/o function variant is completed immediately upon its initiation, a skip over the next instruction may be caused by the peripheral controller.

It is programming error to initiate an i/o function that has already been initiated by pause mode or break mode iof instruction, but has not yet completed. No diagnostic hardware is provided.

In time-shared operation, a computation running in one or more fields of core memory may consist of several or many processes that are separately scheduled by the system. These processes may execute i/o functions as permitted by the program reference list of their computation. Each process may execute i/o functions independently from the others (subject to the previously mentioned restriction that just one variant of any particular i/o function may be in execution at once). Thus i/o concurrency is achieved through parallel programming and the use of the fork, quit and join meta-instructions explained elsewhere. Sequence break operation is not available for control of i/o functions in time-shared operation. In/out operations are controlled by the ivk instruction (invoke) that has the format shown in Fig. 3a. The 6-bit index selects an entry in the program reference list that has the format of Fig. 3b. This word is established by the system upon request by a computation and is not directly alterable by a computation. The ivk instruction requires two memory cycles, one to fetch the ivk instruction and one to fetch the program reference list entry.

Two modes of i/o function timing control are available according to bit 7 of the ivk instruction.

mode	bit 7	
pause	0	Processing is suspended until the most recent wait mode i/o function has completed. Execution of the current i/o function variant is started. If execution of the i/o function variant is immediately completed execution of this process is continued. Otherwise the process is made dormant by the system until the i/o function done signal is received, whereupon the process is rescheduled by the system. Meanwhile the system may execute other processes of this and other computations.
wait	1	Processing is suspended until the most recent wait mode i/o function has completed. Execution of the current i/o function variant is started and processing is continued whether or not execution of this i/o function is immediately completed. If a wait mode i/o function is not completed within the time limit contained in the reference list entry, a "function tardy" exceptional condition occurs and operation of the process is terminated by the system. The value of the time limit is n x 500 microseconds where n is the value of the 3-bit t field of the reference list entry. No time limit is applied if the u bit is set.

It is a programming error to execute an i/o function that has already been started by a pause mode ivk instruction, but has not yet completed. In such event a "function busy" exceptional condition occurs and operation of the process is terminated by the system.

The I/O Function Bus

This section describes the function of the 36 lines comprising the i/o function bus, and the signaling conventions used to communicate i/o function initiation and completion between a peripheral controller and the PDP-1-X. Fig. 4 identifies the lines. For all lines, a value of one is represented by ground and zero by minus three (nominal) volts. All lines are series terminated at the PDP-1-X and are intended to be left open at the free end of the i/o function bus. Some lines are driven only by the PDP-1-X namely

FN[6] FV[4], FTM, FSV, FCN

The latter three signals control the timing of all communications over the bus. Lines FPZ, FSK, FDN, and FRQ are driven only by peripheral controllers. The signal on one of these lines has value 1 if it is being driven to 1 by any peripheral, that is, the signal seen by the PDP-1-X on these lines is the logical or of the signals applied by the peripheral controllers. The lines FDA[18] are used in both directions. They are similar to the inward lines with the addition that the PDP-1-X may also apply a signal. All units are also able to sense the signal value on these lines. Each line is used to transmit signal in one direction at any time. The "oring" property of lines used to send signals to the PDP-1-X is used in the case of the service request line FRQ which indicates that one or more peripheral controllers desire to send i/o function completion information to the PDP-1-X. The other lines

(with the exception of data lines FDA[0-7] which double to send in priority signals) are restricted to use by the peripheral controller selected by the six FN lines.

The lines of the i/o function bus are time-shared by two activities:

a) initiation of i/o function variants upon signal from the PDP-1-X.

b) transmission of completion signals to the PDP-1-X upon signal from any peripheral.

We describe the signalling conventions for these two activities separately.

The I/O Function Cycle

Execution of an iof variant requires transmission of 18-bit data words to and from the peripheral, according to the variant and function numbers supplied. The PDP-1-X must find from the peripheral whether the i/o function is immediately complete, and, if so, whether a skip of the following instruction is indicated. The timing of this activity is controlled by lines FCN and FTM as shown in Fig. 5.

A function cycle consists of 2, 4, 6, or 8 two-microsecond time intervals as indicated in Fig. 5. The number of intervals depends on the amount of information to be exchanged between the peripheral controller and the PDP-1-X. In response to an iof or ivk instruction, the appropriate variant and function number are applied to the FV and FN lines, and a function cycle is begun. The FV and FN lines are held static throughout the function cycle. The signals applied by the selected peripheral controller to lines FDN, FPZ, and FSK are examined just before the end of intervals B, D, F and H to determine subsequent action. If line FDN has value 0, then the following interval (C, E, or G) is used for an information transfer via the FDA lines according to the following schedule.

	line value		
	FPZ	FSK	information transfer
FDN = 0	0	0	no action
	0	1	IO register of PDP-1-X to peripheral controller
	1	0	AC register of PDP-1-X to peripheral controller
	1	1	Peripheral controller to IO register of PDP-1-X

During timing intervals C, E, and G, the unit which is the source of the data being transferred applies the data signals by shorting to ground (with a series termination of 82 ohms) the lines that are to have value one. Outside intervals C, E, and G the PDP-1-X will reset the FDA lines to minus three volts to insure fast settling of these lines for the next data exchange. Lines FDN, FPZ, and FSK are reset to minus three volts by the PDP-1-X outside timing intervals B, D, F and H.

If the peripheral controller responsible for the selected i/o function applies value one to line FDN during interval B, D, F, or H, the function cycle terminates at the end of that interval with the following interpretation by the PDP-1-X.

	line value		
	FPZ	FSK	interpretation
FDN = 1	0	0	execution of i/o function complete; normal continuation of process.
	0	1	i/o function started; process execution suspended.
	1	0	execution of i/o function complete; process continues with skip.
	1	1	i/o function previously initiated and not complete.

Function Service Cycle

Proper operation of the PDP-1-X system depends on the correct identification and acknowledgement of each i/o function completion signal. This is accomplished over the i/o function bus by the performance by the PDP-1-X of a service cycle whenever a signal of value one is present on the request line FRQ and a function cycle is not in progress. The timing of a service cycle is controlled by lines FSV and FTM as shown in Fig. 6. The service cycle consists of five two-microsecond intervals A, B, C, D and E. During the service cycle each peripheral controller which had applied a request signal to the FRQ line prior to the beginning of the service cycle participates in a cacus to determine the number of the i/o function whose completion is acknowledged by this service cycle. The function chosen is the i/o function having the highest number among those in the cacus. During interval |A| each participant in the cacus applies a one to the line among FDA[0 - 7] corresponding to the first octal digit of its i/o function number. At the end of interval |A| the PDP-1-X determines the highest numbered line having value one among FDA[0-7] and applies the corresponding octal digit to lines FN[0-2]. Those participants in the cacus whose function numbers do not match FN[0-5] in the first octal digit leave the cacus at the end of interval |B|. During

interval C each participant remaining the caucus applies a one to the line among FDA[0-7] corresponding to the second octal digit of its i/o function number. At the end of interval C the PDP-1-X again determines the highest numbered line having value one and applies the corresponding octal digit to lines FN[3-5]. Lines FN[0-5] now hold the number of the i/o function whose completion is acknowledged by this service cycle. The appropriate peripheral controller must reset the completion indicator for this i/o function at the end of time interval D. The controller responsible for the acknowledged i/o function completion signal must remove its signal from the request line FRQ at the end of time interval D to insure that a spurious service cycle does not occur.

Timing Example

Some simplified examples of timing are shown in Figs. 7, 8, and 9 for time-shared operation. In Fig. 7 the simplest possible case is shown. The execution is immediately complete without a skip, no data exchange is involved, and not wait mode i/o function remains incomplete. The time required is two memory cycles at 5 microseconds each and a function cycle of 4 microseconds duration or a total of 14 microseconds. In Fig. 8 a wait mode i/o function is initiated at a time when no previous wait mode function is incomplete. A function cycle initiates execution of the desired variant and process execution continues. Upon the next invoke instruction execution is suspended until the completion signal has been received. The timing is apparent from the diagram. In the function cycle shown a data word is transferred to the peripheral controller from the PDP-1-X AC and a data word is read into the PDP-1-X IO.

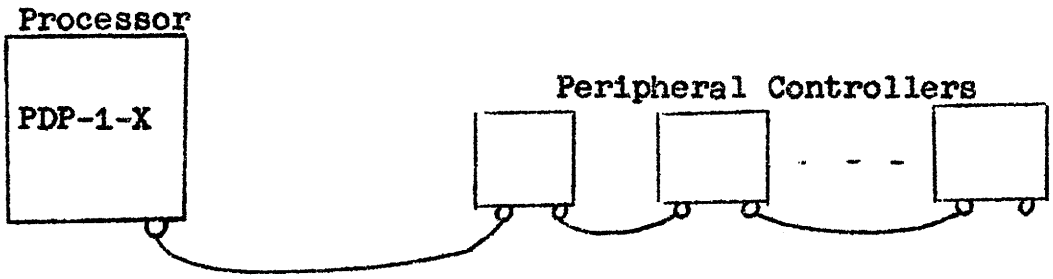


Fig. 1 -- I/O Function Bus

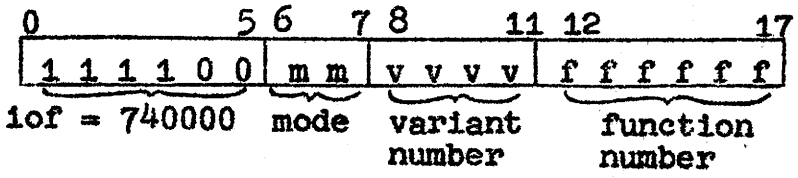
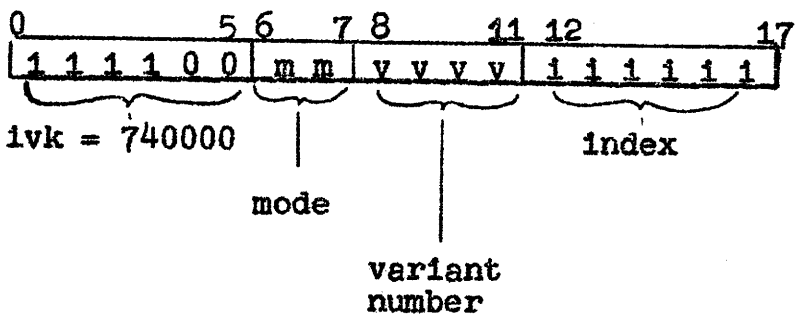


Fig. 2 -- Format of iof Instruction

a)



b)

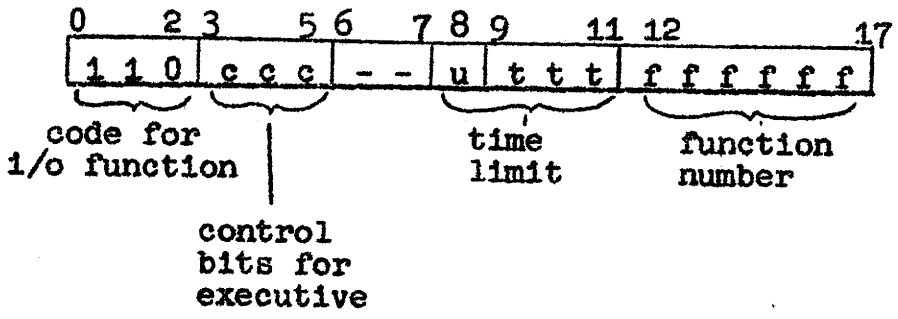


Fig. 3 -- Format of ivk instruction and i/o function item in program reference list.

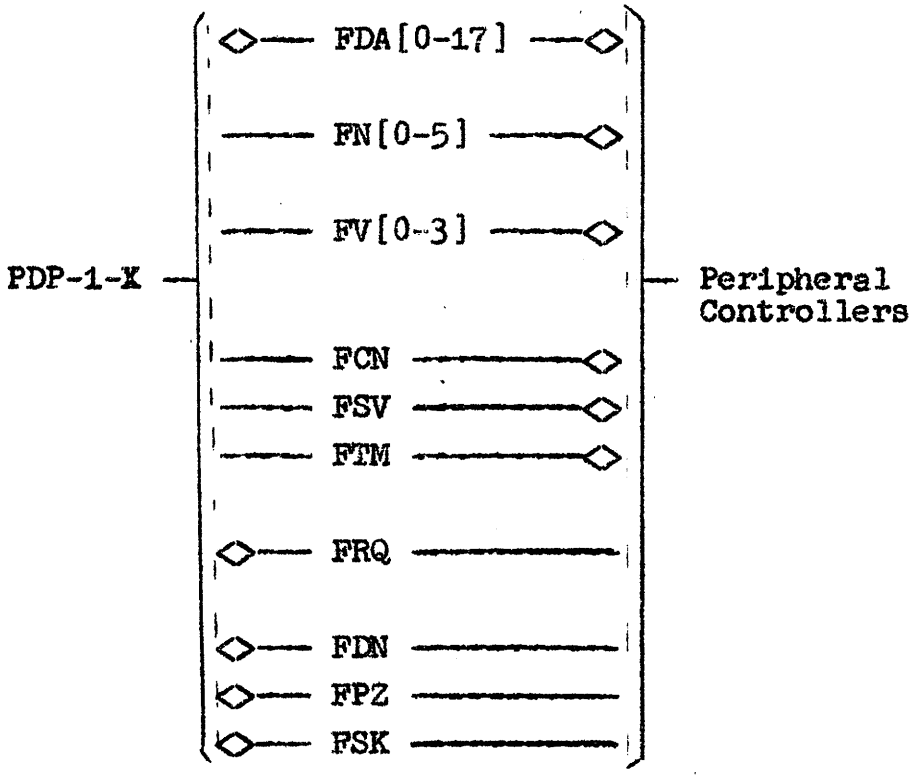
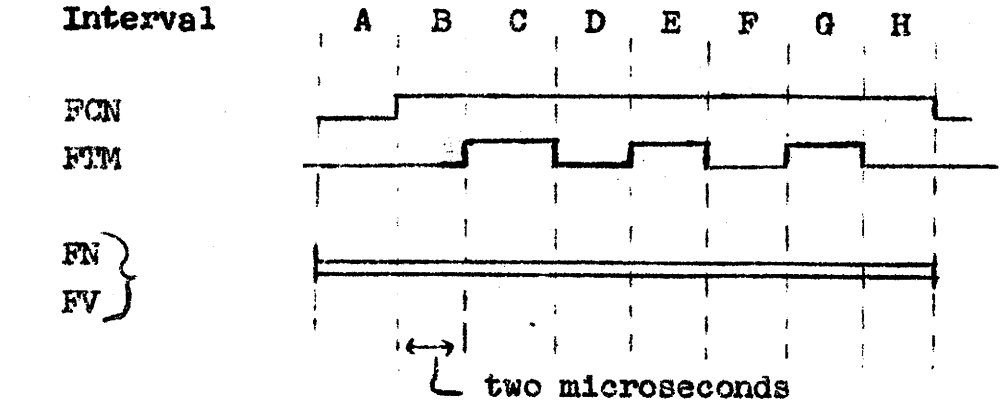


Fig. 4 -- Lines of the I/O Function Bus

a)



b)

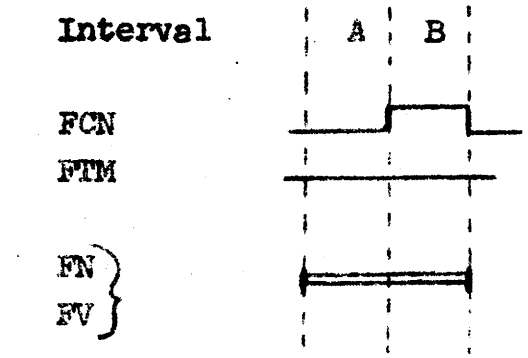


Fig. 5 -- I/O Function Bus Timing Cycle of
a) 8 intervals and
b) two intervals

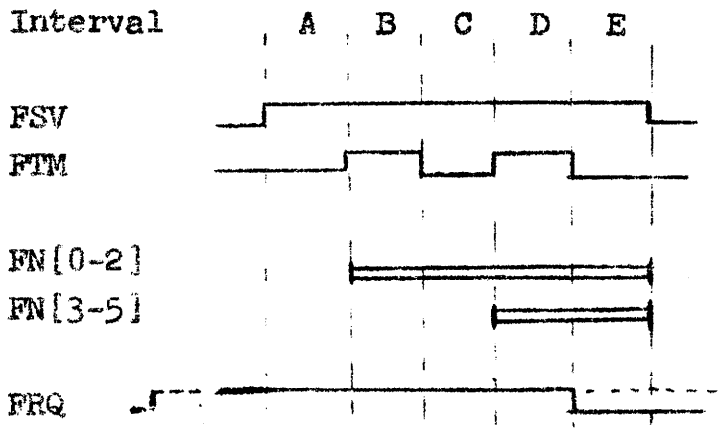


Fig. 6 -- Timing of Function Service Cycle

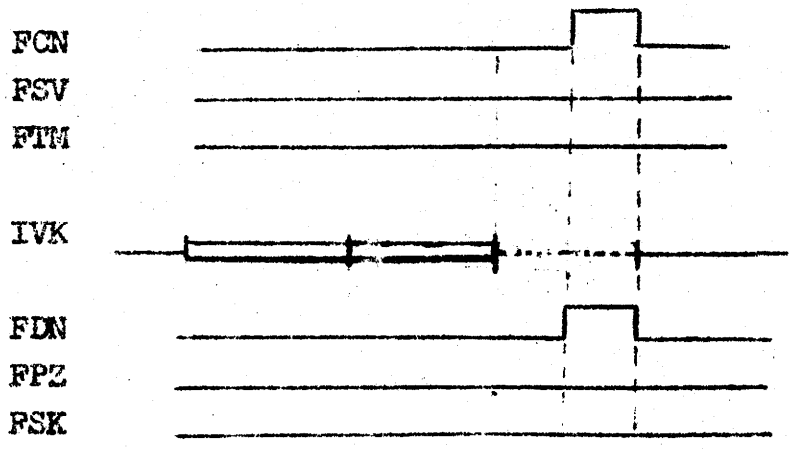


Fig. 7 -- Simplest iof Execution Timing

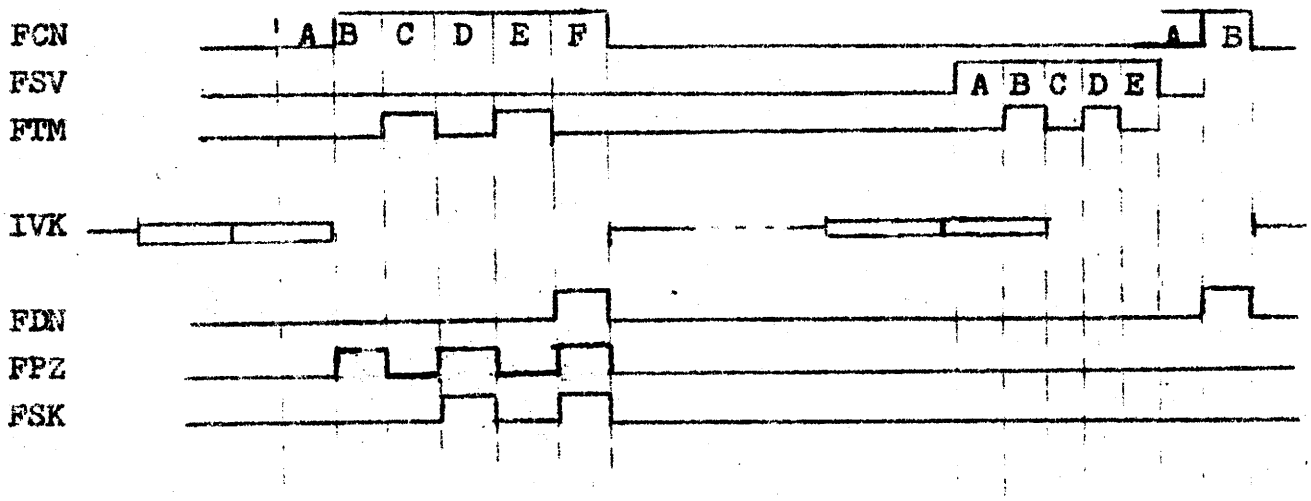


Fig. 8 -- Timing of a Wait Mode iof Execution