PERMANENT MEMORANDUM NO. M-1110

SUBJECT: Operation of PDP-1 Sequence Break System and High Speed In-Out Channel for Effecting System Intercommunication

DATE: June 20, 1961

TO: PDP Distribution List

FROM: Computer Division

Normal In-Out Intercommunication

Where several devices will be communicating in random fashion with the computer, a means is required to signal state of readiness to transfer information and to synchronize this transfer with PDP-1 machine timing.

The PDP-1 has program flag flip-flops which can be set by external ready signals and sensed by the program. If the computer were to discontinue a main sequence until it could obtain another word from a device, the following program steps could be used:

```
x       szf k       skip if k is not set
        jmp         (jump to in-out subroutine)
        jmp x
```

The machine would continually check the flag until its readiness condition set the flag. It would then jump to an appropriate iot instruction and proceed to another wait or to the main sequence. If it is desired that the main program not wait for the device, it would include periodic sampling of the flag by szf and jmp to the in-out subroutine instructions.

The limitations of the above technique are clearly:

1. Operating time is consumed in sensing and waiting by the program.

2. Priorities among several devices must be established by the program in its selection of program flags. This often requires careful timing considerations by the programmer and is a particular limitation when slow and fast devices are concurrently requiring communication on a random basis.

Alternately, flip-flop states (in the special equipment) may be read into the In-Out Register using iot instructions. The state of the IO can then be checked with program logic.

Sequence Break System

The Sequence Break System is an optional in-out control available for PDP-1 which allows concurrent operation of several in-out devices and their related subroutines together with the main program sequence.

A system is then desirable for enabling a number of devices to communicate in random (asynchronous) fashion with the computer with better utilization of computer speed.

The Sequence Break is such a system and basically enables:

1. Automatic interruption of the program and break to a sequence when initiated by the completion of an in-out device, by an external signal (such as from a filled IO device buffer), or by the program.

2. Interruption on a priority based on the speed of the device concerned with respect to the other devices. Magnetic tape, for example, would have high priority with respect to typewriter. This enables the faster device to interrupt the slower sequence effecting concurrent operation of both devices.

The system utilizes the PDP-1 in-out register for transfers and nominally has 16 channels arranged in a priority chain. The readiness state signal from each device is wired into one of these automatic interrupt channels. Any device, with appropriate buffering, may be connected. A typewriter would, for example, normally use two channels, one for entry and one for readout. The channels themselves are basically an arrangement of control flip-flops.

Each channel has four major memory registers permanently assigned to it. When an active channel assumes priority, the contents of the live registers (AC, IO, and PC) are first stored in these locations. The fourth register contains an instruction which is usually a jump to a suitable routine.

When a sequence break channel causes an interrupt, program control is transferred to this fourth register. The program then is operating in the new in-out sequence. A typical program loop for performing the in-out sequence would contain three to five instructions, including the appropriate iot. This in-out sequence is usually followed by a load AC and load IO from the fixed locations and a special indirect jump through the location of the previous C(PC). This last instruction terminates the sequence by turning off the channel flag for this sequence.

The Sequence Break instructions are, then:

```
Automatically store AC, IO, PC   }
jmp    jump to sequence          }  20 usec
 .
 .
iot                              }
 .                               }  In-Out Sequence
 .                               }
lac    restore registers         }
lio                              }  Terminating Se-
jmp*   channel number times 4    }  quence 30 usec
```

Time to enter a sequence is 20 usec. Time to restore is 30 usec. The sum of these plus sequence time is the total time required to perform the transfer for that channel.

The in-out sequence may transfer a number of words through a channel and may itself be interrupted by a higher priority device. In this case, the current step in the in-out sequence would also be stored and the sequence restored when it regained its priority.

High Speed In-Out Channel

The high speed in-out channel or PDP-1 is an optional feature used to transfer blocks of words between memory and an in-out device, usually a high speed device such as magnetic tape.

A common in-out transfer requirement is the transfer of successive in-out words to or from the computer at the speed of the in-out device. The use of program flags or automatic interruption of the main program by the Sequence Break to perform the transfers will be efficient of computer time when the device speed (or rate of transfer) is slow with respect to the computer operating speed. For most devices, the number of program steps required to transfer each word is small with respect to the number of program steps over the whole period between transfers.

For high speed devices, such as magnetic tape, the reduction in available time between transfers makes it desirable to reduce the transfer sequence time and improve the percentage of computation time to total time computer time.

The high speed in-out channel reduces transfer time to a single machine cycle (5 usec.) Following program selection of the in-out device and definition of the character of the transfer, control passes to the channel for transferring successive words.

When wired to this channel, a device communicates directly with memory through the Memory Buffer Register, bypassing the In-Out Register. To start a device and block transfer, the in-out register is loaded with the memory address of the first word to be transferred, and the accumulator loaded with the last address of the word block to be transferred. The appropriate device is then directed to the proper mode of operation initiated by an iot command. The time required to initiate the transfer is 25 microseconds (load IO, load AC and iot). The entire block transfer then proceeds without disturbing the main program.

The channel is automatically interrogated at the completion of each machine instruction. If it has a word for, or needs a word from, the memory, the current program sequence pauses for one memory cycle in order to serve that channel. Completion of the full transfer will either set a program flag or, when sequence break is connected signal the proper sequence.

There are three registers in the channel. An 18 bit buffer, the current address counter, and a static last address buffer. The latter two are given the information from the in-out register and indexed to perform control of the block transfer.

Using sequence break, but not the high speed channel, to transfer a block of data from magnetic tape, approximately 80 microseconds would be required for each word transfer (50 for a sequence break and 30 to handle data transfer). With 400 microseconds between word transfers, approximately 80% of computer time is available for computation.

Without sequence break or high speed channel, approximately the same time is available between transfers but normally is not useful for programming. This is true because continued interruption of a main program for relatively few operations is inefficient. Without sequence break, therefore, effective computation time is nil. An important exception would be a program which could utilize this computation time between transfers for operations on the magnetic tape data, such as format control, etc.

With only the high speed channel, available computation time is 395 microseconds or 98.7%. The advantage of the channel would be even greater since the above comparison ignores time consumed in beginning the block transfer.

It is clear that the advantage of the high speed channel decreases sharply with slower in-out devices but may be highly desirable for use with magnetic tape.