THE FASBAC REMOTE ACCESS SYSTEM

Dan W. Scott
Vice-President, Research and Development Division
Computer Utility Network
University Computing Company
1949 North Stemmons Freeway
Dallas, Texas 75207

14 April 1969

This is a slightly revised edition of the previous paper dated 13 May 1968

SUMMARY

FASBAC is a conversational system which facilitates setting up runs for the University Computing Company Direct Access Computer utility, via remote low speed terminals in the customers' offices. The system provides processing which is characterized as "remotely initiated batch processing". It is also the basis for a general-purpose direct access file system.

Successful Time Sharing.

Time sharing has become within the computing world the subject of lengthy debates on milli-second response time, swapping strategies, exotic memory management techniques, and whether bits should be colored blue or green. Were these truly critical parameters, the world would still be waiting for its first commercial quick response remote access system. Fortunately these are not the real issues, since time sharing has become highly successful with what are regarded as primitive systems. It would be well to examine the reasons behind the success.

First, it should be noted that time sharing and batch processing systems have two things in common: they use essentially the same kind of hardware, and they are both shared by many users.

Unlike batch processing, however, time sharing has brought convenience and simplicity to computing.

The successful time sharing system differs from the typical batch processing system in four significant areas.

1. It offers convenient physical access.
2. It offers quicker response.
3. It offers simple communication in both programming and control languages.
4. It offers program and data storage and editing facilities.

Simple communication with the computational facility is obviously attractive to the commercial user of calculational services; but, historically, progress in making general purpose systems easy to use was relegated to a secondary priority for many years after the appearance of FORTRAN. However, in specialized applications areas such as numerical control, report generation, CPM/PERT, and circuit analysis, new languages reduced the communication problems between the user and the computer. Finally, JOSS, developed by the RAND Corporation, and BASIC, developed by Professor Kemeny of Dartmouth College, appeared. Both the JOSS and the Dartmouth BASIC Systems reduced greatly the impedance mismatch between programming systems and the general engineering user.

These systems demonstrated that an acceptable commercial system must not only be physically convenient, but must be intellectually approachable as well. A sense that the machine is not critical is important; these direct access systems appealed to many customers because errors are more rapidly corrected in privacy; a mistake is pointed out by the computer with a slap on the wrist, instead of the face.

As important as well thought-out programming languages, are simplified, well thought-out text editors and file systems. This element is not even now widely appreciated: the advantage to a user of ready access to calculational power, and, at the same time, of ready access to the power of computer-aided program filing, data filing, and editing. This file manipulation facility is the very foundation which makes simplicity technically attainable in the successful general purpose, remote access, computation system.

So, people are willing to pay for access to computational facilities. But this access must be both physically and intellectually convenient, and must apply as much as possible to all aspects of the computer utility business--commercial as well as technical, filing as well as computational. In summary, these requirements must be satisfied: reasonable programming languages, reasonable file languages, convenient access to the facilities, and for many applications, quick response.

<u>The University Computing Company Computer Utility Design</u>.

There are several approaches to remote computer sharing design. On the one hand, we have the systems typified by the GE-265, the Dartmouth GE-625, and the SDS-940 small-scale time sharing systems. What these facilities do, they do well; however, they are of limited value to many potential users because of the limited resources allocated to each user. The objective of other designs has been to maximize the usefulness of a large processor. The central processor and its peripheral storage is intended to be fully available to each remote user, at the same time that it is carrying on conversations with all users. But the centralized approach has two drawbacks: (1) Complex and expensive implementation; (2) Poor cost/performance ratio.

Our approach with the UCC FASBAC design is to decentralize the large-scale processing; to distribute functions among subsystems; to be eclectic, selecting small, slow, processors and memories to do those things they do more economically, to allow the user of the large processor its unrestricted power. We combine the two implementation approaches just mentioned.

The UCC direct access design comprises three independent subsystems; the large-scale UNIVAC 1108 subsystem for compilation and numerical computation, the COPE remote terminal subsystem for high-speed card input and print output, and the FASBAC remote terminal subsystem for conversational input and output editing. The UCC Direct Access system might well be called a large-scale time sharing system.

Two of these subsystems have required major development efforts by the University Computing Company for their creation, and all require substantial continuing development.

In order to supply physical access for large volume work, we have put high-speed card reader and printer terminals as close as possible to the user: this is our COPE communications facility. Second, we are developing a low volume communications facility, FASBAC, for file manipulation and text editing. This conversational text editing facility uses low to medium speed terminals in the users' offices and mass storage at the UCC utility center, and is connectable on demand to the 1108 and to the high-speed terminals.

The COPE subsystem multiplexes remote high-speed card readers and printers into the UNIVAC 1108 (Fig. 1). Figure 1 is simplified, and does not show, for example, the COPE communications controller. Both this central multiplexor and the remote terminal controllers use Digital Equipment Corporation PDP-8 processors. The COPE development is proprietary to UCC, but was a natural outgrowth of the UNIVAC 1107 EXEC-2 remote 1004 facility. However, quantitatively, its ratio of performance to cost is much better than that of the 1004 remote card reader and printer. COPE provides unrestricted access to the large-scale 1108 when masses of data are to be moved. The very speed of the data transfer inhibits, of course, meaningful dialogue with a person.

The FASBAC Subsystem Design.

As we have just seen, COPE facilitates bulk input and output of files by readers and printers which are close to the customer. However, file editing, output file browsing, input file corrections--the low volume, conversational elements of accomplishing jobs--are done via the FASBAC teletypewriter terminals.

FASBAC has as its objectives the increase in usefulness of the powerful UNIVAC 1108 systems to the low volume conversational user, and the provision of a basis for immediate remote access to bulk storage.

Technically, FASBAC is simply a communications-oriented text editing facility with bulk storage (Fig. 2); that is, it has a file system for mass storage, it has editing programs, and these facilities are conveniently available to individual customers through remotely located terminals which time-share a small central processor.

Text can be entered, stored, retrieved, and manipulated via low-cost, low-speed terminals suitable for text entry and display. This text may also be input to or output from the UNIVAC 1108 processor. The processing by the 1108 is then in its normal batch mode of scheduling, with multi-priority gueueing. This method of processing may be characterized as remotely initiated batch processing. To the 1108, the PDP-9 processor looks like a card reader and printer.

In addition, FASBAC terminals can indirectly access the COPE terminals through the 1108 (Fig. 3).

The constraints of the FASBAC subsystem must be observed: low-volume input, low-volume output, and low-volume file processing (even though the files themselves can be large). Because of the relatively small memory of the PDP-9, quicker response to sequential processing of large files is obtained when the processing is done by the large-scale machine, rather than the small-scale processor.

The first objective of FASBAC is to provide convenience and quick response to the large-scale computer programmer. The files which he manipulates via FASBAC in a conversational mode are program files and data files to be executed by the large-scale processor. Formerly, users had to edit or control the editing of their files by means of physical manipulation of punched cards. FASBAC automates these procedures of entering and updating, doing away with the constraints of physical unit records, and making the file editing facilities context-oriented. The user then has personal access to the UNIVAC 1108 without an operator's being required for the mechanics of program and data entry.

The text editing programs include both line-replacement methods, as used in JOSS and BASIC, and context-replacement methods, as used in the MIT, DEC, and other editors. In addition, CASH, a re-entrant resident interpretive program, is used for general purpose text manipulation. The choice of methods is left to the preference of the user.

The file system is straightforward, like that of BASIC, from the viewpoint of file naming, but includes more features to control access. An unusual feature allows efficient use of a file both for sequential and random access purposes.

Now for a description of the implementation details of these FASBAC features.

In contrast to the centralized approach, the FASBAC design has a hierarchy of processors. The control of the conversational complex is the operating system and the programs executed in the

PDP-9. All the seemingly intelligent and friendly conversational aspects of character and file manipulation are carried out by the small processors, not the 1108.

The PDP-9 subsystem is a large configuration for its model. It has 32,000 words of 1-microsecond 18-bit memory, all processor options, a half-million word drum with 8.65 millisecond average latency, a 1,000 card per minute reader, a 600 line per minute ASCII printer, 3 DEC magnetic tapes, and high-speed paper tape I/O. The PDP-9 is the heart of the conversational system. It directs all functions and controls directly all I/O devices except those low-speed devices connected to the PDP-8.

The PDP-8 supplies the low-speed communications interfaces. It is connected to the PDP-9 through a high-speed data channel. Each PDP-8 can handle the equivalent of 32 teletypewriter circuits, each 110 baud, or any of three other speeds. It has no line bit buffers, and samples each line under program control at 8 times the line bit rate. This feature allows a variety of data formats and clock rates. The PDP-8 also does character transliteration, monitors the lines, answers the phone, hangs up, and dials out. The drum buffering of the messages is done by the PDP-9. A PDP-8 can also be located remotely to the PDP-9, and used as a line concentrator, using a duplex voice circuit.

ASCII character codes are used in the file system and from the teletypewriter terminals through the PDP-9 up to the UNIVAC 1108 interface. The UNIVAC 1108 software presently requires six-bit Fieldata code.

The mass storage device presently is the UNIVAC FASTRAND II drum. A multi-access controller, which also does drum address translation, was developed by the University Computing Company and Weismantel Associates. This controller allows the UNIVAC 1108 and four PDP-9 processors to share access to the FASTRAND II drum. It also provides separate duplex paths for core-to-core processor communication, between the 1108 and four PDP-9's. The PDP-9 time-sharing file system is completely separate from that of the large-scale 1108 processor. We do not allow the large-scale processor to access the files of the time-sharing system.

The UNIVAC 1108 subsystem is reserved for the computational muscle of the complex. From the viewpoint of the PDP-9, it is used as a peripheral device, for unrestricted batch queued work on the 1108. Note that the conversational user enters jobs only indirectly into the 1108, via the PDP-9. This technique is described as remotely initiated batch processing.

Hardware Selection.

Time-sharing hardware is essentially similar to batch processing hardware. There are certain differences of emphasis. In addition to general criteria, such as anticipated reliability, maintainability, and satisfactory delivery schedule, the cost of the system must be evaluated in terms of other criteria regarding performance:

First, the processor must have a good means of handling I/O. For example, the interrupt scheme must be efficient, and allow multiple, dynamically changeable priorities. A variety of various-speed devices must be attachable at a cost, in each case, appropriate to the device's requirements. These devices range from bulk store through swapping drums to large numbers of slow-speed terminals of various speeds and line disciplines.

Secondly, a variety of communications-oriented interfaces must exist for the subsystem. The low-speed modem control must handle dial-in and dial-out, as well as private-wire service. There must be high-speed modem interfaces.

The processor and memory word size must handle efficiently characters of at least seven bits each, as this size character is better for conversational terminals than the six-bit character. The choice of word size is also dictated by the requirements of interfacing with the 36-bit UNIVAC 1108.

Finally, the designer of the conversational processor must have made the appropriate economic trade-offs, for a given total cost, between processor and memory speed on the one hand, and elaborateness of the instruction repertoire on the other hand. If there are to be many simultaneous users, considerable memory is required; but the elementary operations performed by the processor are rather simple-minded. Therefore, it is essential that the designer of an economic processor not sacrifice memory size and speed for an elaborate instruction set.

In summary, the criteria for the FASBAC processor hardware involved the exercise of judgment to form an opinion regarding the final cost/performance ratio of a configuration, as of a certain date, but using different criteria than the usual ones of arithmetic capability. Needless to say, the ideal hardware was not found.

Conclusions.

The major engineering advantage of the UCC FASBAC design concept is the decoupling of many of the conversational elements of remote access from the highly efficient batch processing 1108 and the COPE high-speed card readers and printers. This master/slave design approach has been used in about 80% of the successful general purpose time-sharing systems of the past. We are just pushing the concept a little. We feel that this decoupling from the large, expensive, processor is the only economically rational approach to give conversational access to a major computer such as the UNIVAC 1108. Quick reaction to a keystroke is not economical with the same expensive processor and memory that can also invert a hundred by a hundred matrix in a few seconds. The hierarchy of processors and memories is a present-day economic necessity.

It has been demonstrated in the software area that several stages of impedance matching are necessary between the user and the computer system. The UCC Direct Access utility utilizes stages of impedance matching in order to handle efficiently the wide range of computer applications which today's users create.

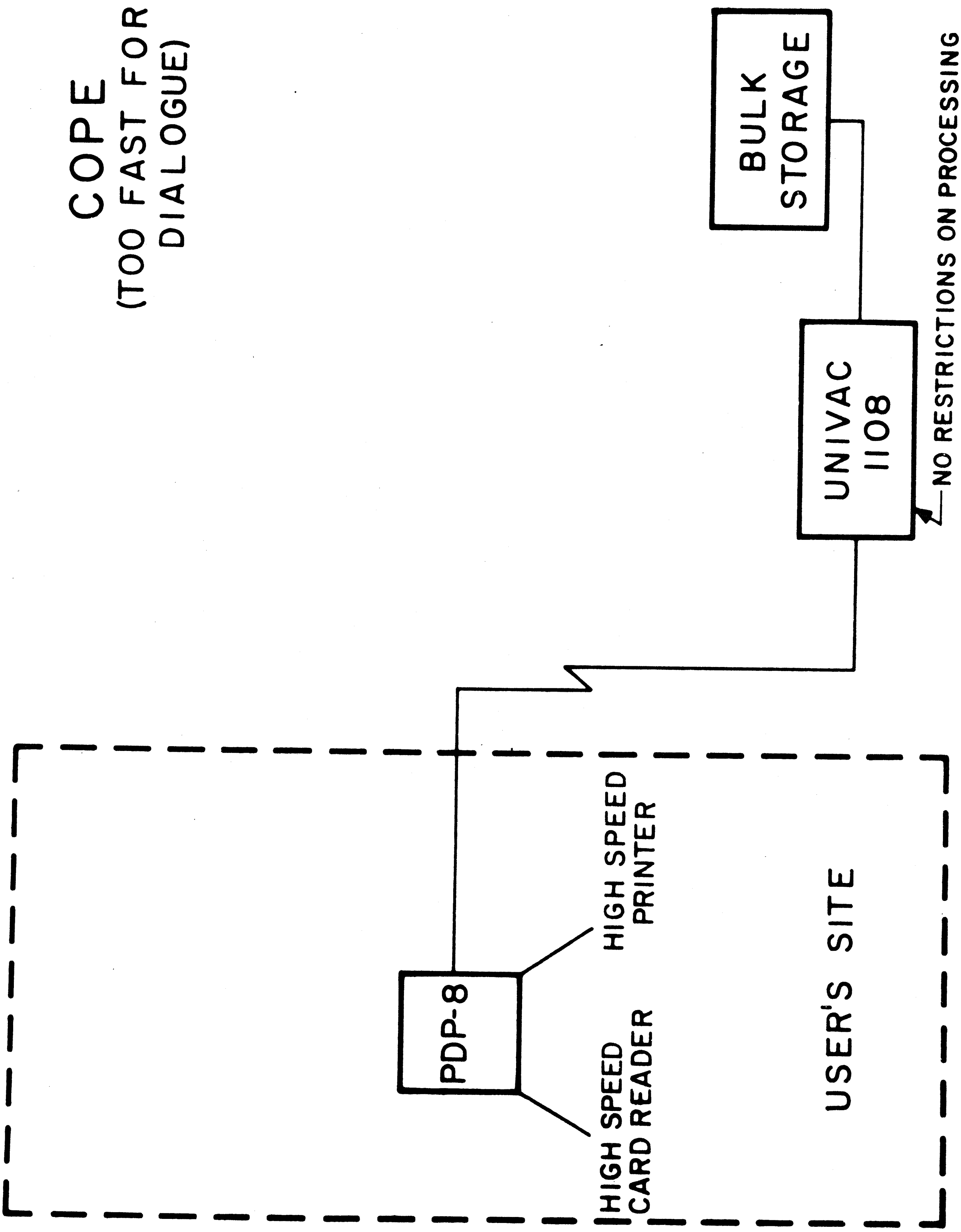

FIGURE 1. COPE

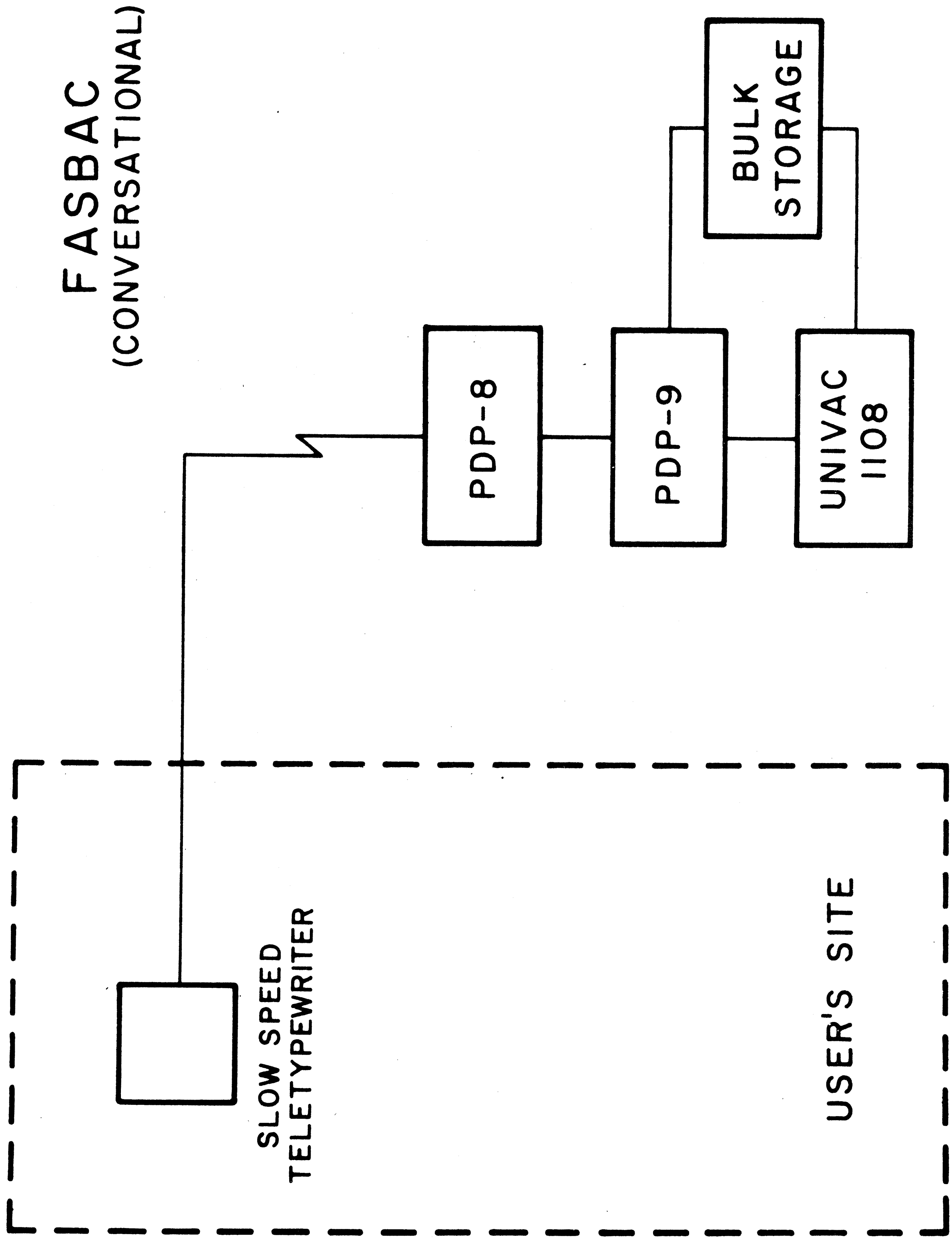

FIGURE 2. FASBAC

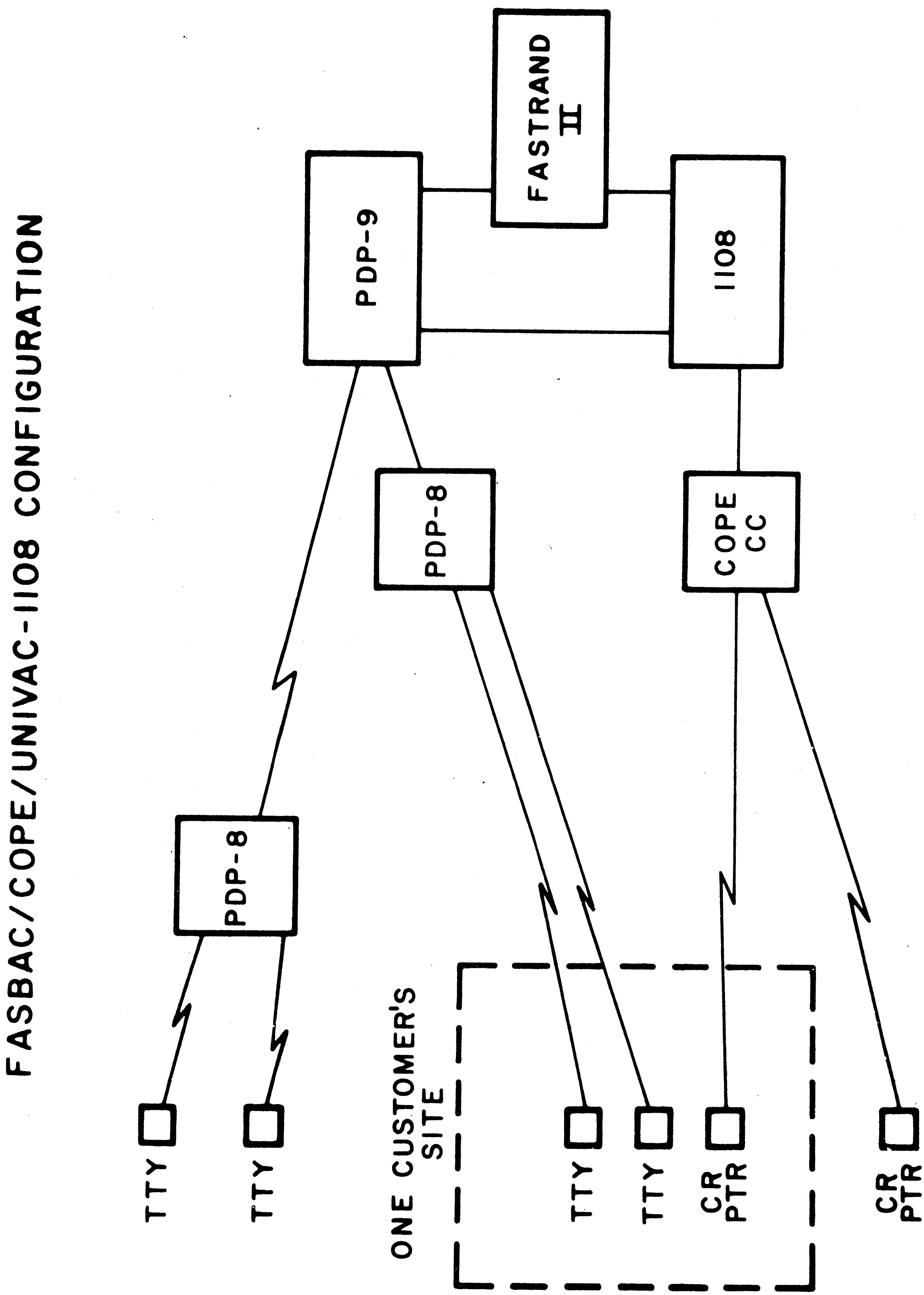

FIGURE 3. THE UCC DIRECT-ACCESS COMPUTER UTILITY