Title
ERROR CHECKING AND OTHER ASPECTS OF DATA-LINK ORGANIZATION

Author
Lafore, Robert W.

Publication Date
1967-11-08

University of California

Ernest O. Lawrence Radiation Laboratory

ERROR CHECKING AND OTHER ASPECTS OF DATA-LINK ORGANIZATION

Robert W. Lafore

November 8, 1967

Berkeley, California

Paper presented at the Fall DECUS
Symposium, Anaheim, California,
November 10 and 11, 1967.

UCRL-17934
Preprint

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

ERROR CHECKING AND OTHER ASPECTS OF DATA-LINK ORGANIZATION

Robert W. Lafore

November 8, 1967

ERROR CHECKING AND OTHER ASPECTS OF DATA-LINK ORGANIZATION*

Robert W. Lafore

Lawrence Radiation Laboratory
University of California
Berkeley, California

Abstract

The arrangement of registers in the data link permits error detection and correction and solves the problem of synchronizing the computers at the ends of the line. Preliminary information must be exchanged between the two synchronizers before a data record may be sent. The approach to the logic design of the system is also briefly described.

Introduction

The organization of the Data-Link was conceived with two primary requirements in mind. First, a powerful error correction and detection system is necessary, and second, a means for the computer programs to communicate information concerning the data record to be sent must be provided. This paper describes the methods used to achieve these goals.

Link "Conversation"

Before the data can be transferred via the link, the I/O routines in the computers at each end of the telephone lines must exchange information regarding the format of the data to be transmitted. This is done by nondata words, which, borowing from CDC terminology, are called "function words" if they go from the 6600 to the small computer, and "status words" if they are initiated by the small computer.

An initial status word from the small computer is used to signal the operator of a service request, via the console light, and a function word carries back his push-button response (see Fig. 1). Later, when the operator has loaded the Analysis program, the program itself initiates the sending of a function word to the small computer to inform it that the data may now be transmitted.

Since the Fortran Analysis program in the 6600 requests only logical records of a certain length, the peripheral processor (PP) I/O routine must obtain the physical record length from the small computer: This is sent as a single data word. To read in this word (as to read in any status or data words) the PP first sends a function word, in this case "GO-WORD-COUNT", whose only purpose is to cause the status word to be transmitted from the channel synchronizer (CS) to the PP.

Following the transfer of each physical record, the small computer informs the PP that either (a) more physical records must follow to complete the logical record, in which case an "end of physical record" status word is transmitted, or (b) the logical record has been completed, in which case an "end of logical record" is sent.

Data output (from the 6600) is analogous to data input except that the PP must inform the small computer at the beginning of each record whether the record will be composed of data or only an end of file.

Since the status and function words are to a large extent initiated and interpreted by software, the "elements of the conversation" described above can be altered to meet future changes in the 6600 operating system or even to provide the interface for entirely different computers at either end of the link.

As noted in the previous paper, data is transferred in words of 12 parallel bits. Function and status words are distinguished from data by the presence of an extra bit which is transmitted as the 13th bit

*This work was done under the auspices of the U. S. Atomic Energy Commission.

in parallel with each word. This 13th bit has the logical value "1" when a data word is sent, and "0" when a nondata word is sent. Since every word must contain at least one bit to be recognized by the receiver, the 13th bit also provides a means of recognizing an all-zero data word.

The Carrousel

The four registers connected directly to the long-line receivers and transmitters (see Fig. 2) are called the "carrousel" and constitute the heart of the error checking and correcting system.

During normal data transmission (for example, during input to the 6600), a word flows out of the small computer memory buffer into the B register (BR) in the device synchronizer (DS), then into the CR, and finally the DR, where it is both stored and transmitted to the CS. In the CS it is received in the CR, shifted to the DR, and retransmitted to the DS. Arriving in the CR of the DS, this echo is compared with the original word still in the DR. If the comparison succeeds, both CR and DR are cleared. During the round trip of this first word from DS to CS and back, a second word will have shifted from the small computer memory to the BR, where it awaits a successful comparison of the previous word. When both CR and DR are cleared, this word is free to shift into the CR, and the sequence is repeated again. A similar process takes place for data output from the 6600.

Since I/O operations between the computers and the link are concurrent with the transmission and echoing of the previous word, the cycle times of the computers cease to effect the data rate of the system (assuming line length is the limiting factor). Also, if the computer on the receiving end is not able to accept the data fast enough, the process simply pauses until the received word is finally read in, thus preventing loss of data or the necessity for complex synchronization and timing devices.

Error Correction

If the comparison check on the echoed word should fail, only the CR is cleared, and a special signal called a "fuzzy signal" is sent to the receiving end to warn that the previously received word was in error. This clears the DR in the receiver and permits the same word to be retransmitted as before. The same word may be sent many times until a correct echo is finally received. However, if this echoing process should continue too long, the link assumes that a serious fault has occurred and sends function and status words to this effect to the two I/O programs.

During the periods when the link is waiting for the computers or the 6600 operator to give it further instructions, it arranges that both the synchronizers will be in <u>transmit</u> mode. This ensures that any noise word received at either end will be compared with the zeros in the empty DR and erased, as with an ordinary error. (If the synchronizers were left in <u>receive</u> mode they would retransmit any noise word arriving in them and eventually fill up all registers with replicas of the noise word.)

The "fuzzy signal" referred to above for signalling errors is one of several signals that, because of their critical role in the operation of the link, must be as nearly as possible immune from noise. This is accomplished by sending a pulse train instead of a single pulse as with data or function/status words. Because these signals are very rarely sent in comparison with the number of data words, the additional time needed to provide a pulse train does not increase significantly the time necessary to transmit each record. Other fuzzy signals indicate the beginning and end of a physical record and provide a general reset.

State Diagrams

In the design of the data link considerable use was made of state diagrams to clarify the operation of the system (see Fig. 3). The position of various flip-flops (for example, those indicating whether a particular register is full or empty), constitute "conditions," a particular combination of which generates a "state." The state in turn alters the conditions by performing various operations, e. g., shifting a word from one register to another. These altered conditions then generate a new state, and so forth.

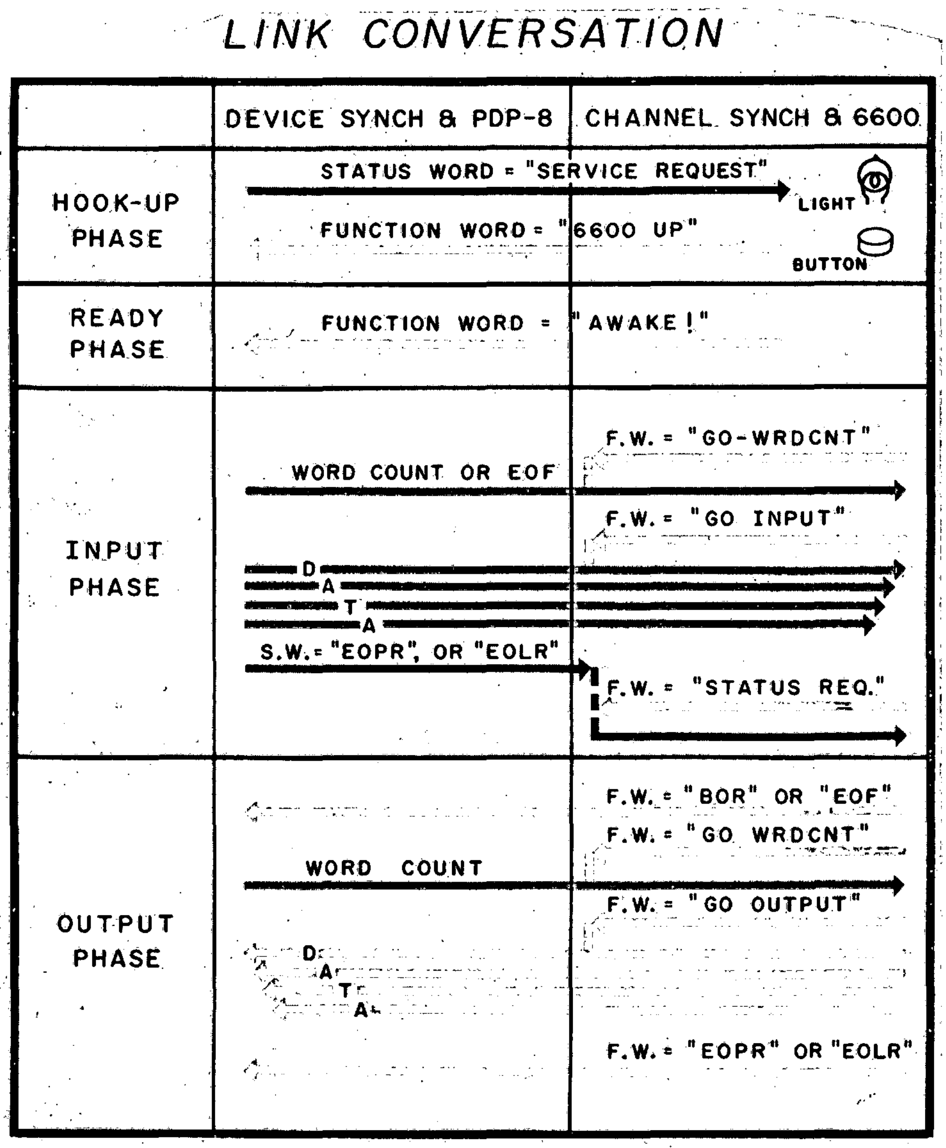
Since each synchronizer contains more than a dozen flip-flops, plus many external signals which also provide conditions, it becomes extremely difficult to keep in mind the exact sequences and timing. Therefore, to completely specify the logical operation of the machine, both for the initial design and for future debugging, a computer program was written to simulate the operation of the logic. This proved invaluable in tracking down some of the more obscure design and wiring errors.

Debug Boards

Debugging of the link required the development of another feature of the system: debug interrupt boards. In normal operation the transitions from one state to another are made very quickly and are frequently nonrepetitive: e. g., a single status word is sent, then the link waits for some response from the outside world. This situation would create extreme difficulties in debugging unless some method of slowing down the transition from state to state could be found. (If this were a clocked system, this effect could of course be achieved by merely slowing down the clock.) The interrupt boards fulfill this function by inserting a switch and a light into the output line from each state. Thus, with the switch open, the conditions for a particular state cause the corresponding light to appear; but the effects of that state will not be enacted until the switch is closed, whereupon the light for the next state in the sequence should appear. Since lights indicating the position of the major flip-flops and the contents of each register appear on the front of the synchronizer control panel, the debug interrupt boards provide a means of seeing the status of all active elements in slow motion. Used in conjunction with the computer printout of the sequence of conditions and states, this provides an easy and effective debugging method.

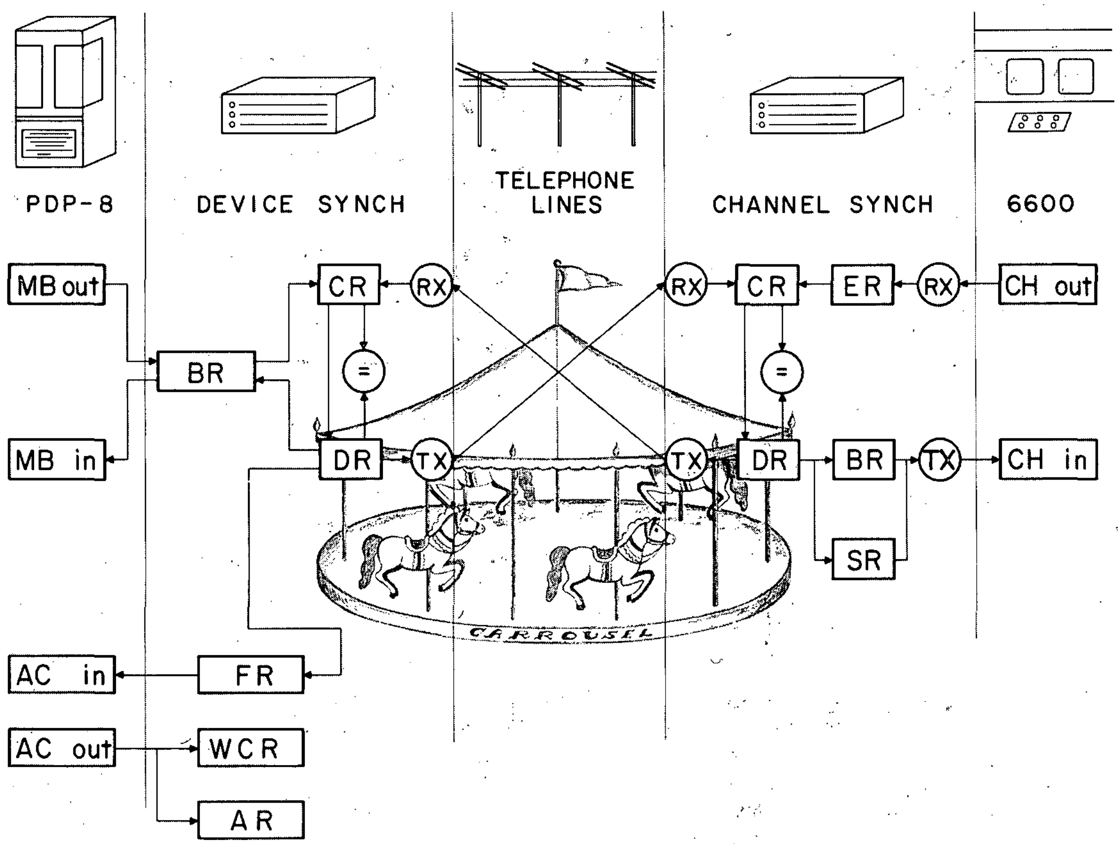
Conclusions

The data-link system provides a reliable and satisfactory solution in its present context to the problem of interfacing small data-gathering computers to a large computer for the analysis of a comparatively small quantity of data at infrequent intervals. Should this type of service become more popular in the future, the link can be readily modified to accept changes in the 6600 operating system (such as using a dedicated PP for all link data) or interfacing to a time-sharing computer devoted to link data handling.



CBB 6710-6240

Fig. 1. Link conversation.



XBB 6710-6260

Fig. 2. Organization of the link registers.

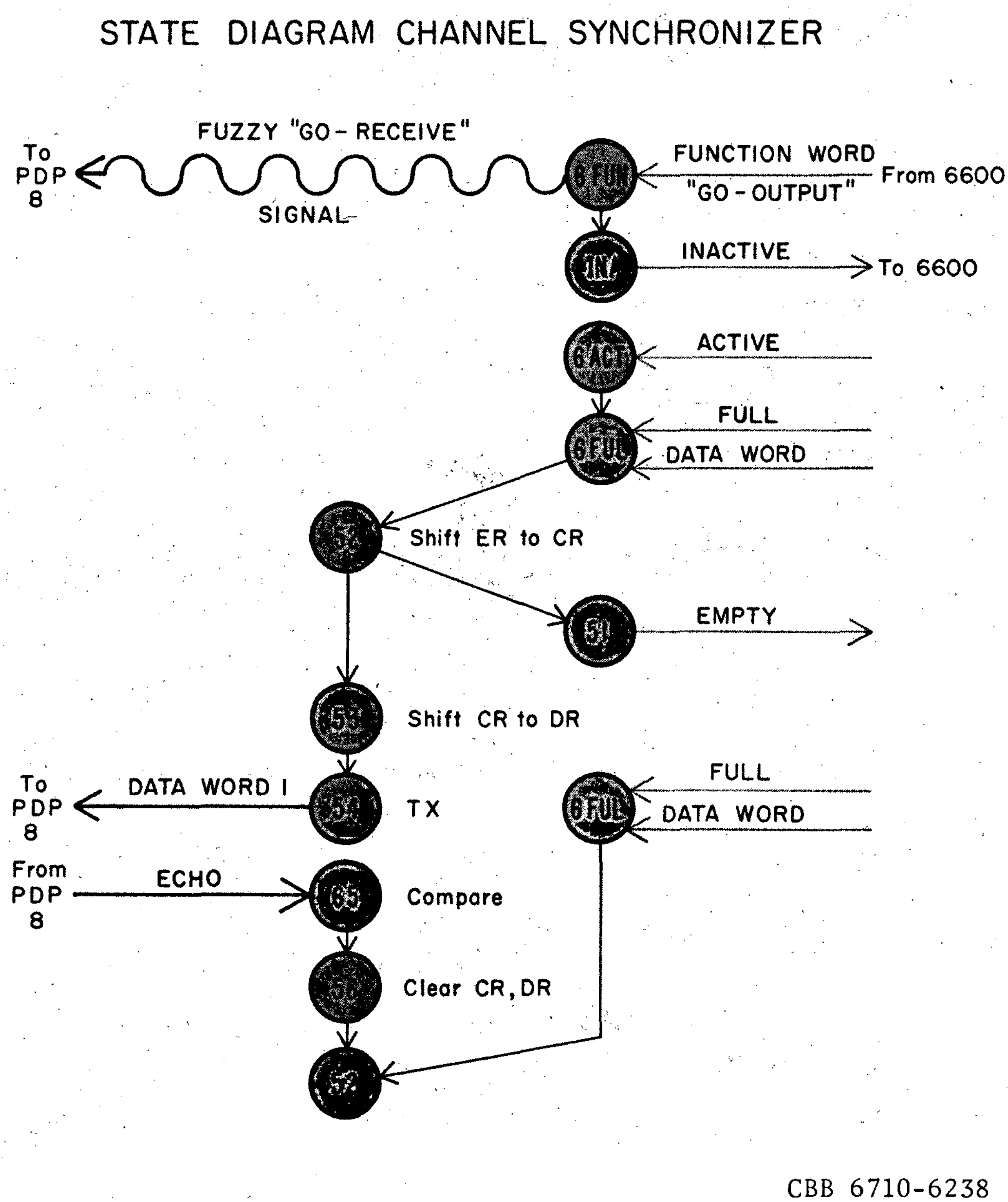


Fig. 3. Typical state diagram (channel synchronizer, data output).