CERN PS/FES/73-1

28.1.1973
jg

AN ALTERNATIVE PROPOSAL FOR THE PS PROCESS-CONTROL COMPUTER LINK

J.H. Cupérus

ABSTRACT

A data link between several process-control computers is described. The link is under control of relatively simple hardware and no special computer is needed to control the link. A priority system assures smooth flow of information, even under peak loads. The link is modular and can easily be adapted to any number of computers. This link could be considered as a possible alternative for the PS multi-computer process-control system.

----

1. INTRODUCTION

The need for a multi-computer network for the PS has been discussed in several notes by J.H.B. Madsen. A specific proposal has already been made by E. Asseo (MPS/CO/Electronique 72/6). I got indirectly interested in this problem a few months ago and the present note is the result of some thinking on the subject. The ideas expressed here are not strictly original, but the purpose of this note is merely to show that an alternative solution could work satisfactorily. Obviously, a choice of the type of link can only be made in view of the real needs, now and in the future, of the PS control system. In a design study for the layout of the future system, and the ways of growing to it, the present link could possibly be considered alongside others.

2. THE NATURE OF THE DATA FLOW

In a multi-computer process-control system, we can distinguish two sorts of data:

(a) real time data, involving communication between a computer and the PS hardware. Communication is established, in general, by the computer which sends out the address of a hardware component and makes a setting or receives the result of a measurement. Communication involves single words at random addresses and a fast access time is often required.

(b) data flow between computers. This communication involves, in general, transfer of blocks of words between fixed parts of the computer memories. These blocks can be either requests for information, well ordered data files or whole program sections. Very fast access time is usually not essential.

These two sets of data are different in nature and content and, while it is certainly possible to construct a link capable of transmitting both, such a link will not be efficient for either set. It is much better to separate the sets as shown on figure 1. All data to and from the PS hardware go directly to the computers. The same information, but now in well-ordered form, can be made available to all computers via the link. This link has a "Dalton"-like structure, with each computer having access to a number of common dataways. The main difference with "Dalton" is that there are no "hardware data" to transmit. The communication is between computers only and, as we shall see, the link can be under simple hardware control, without the need of a special control computer.

As seen from the link, all computers are equal. In practice, however, under software control, one or two computers can take the role of "master" and the other computers will be "satellites".

3. SEND-RECEIVE PROCEDURE

The interface between the link and the computer is shown on figure 2. The registers on the left of the CONTROL UNIT are for sending data to the link and those on the right are for receiving data from the link. To fix the ideas, we suppose that we want to send from computer A to computer B. To distinguish between them, the registers and connections of the receiving computer will be marked with an asterisk (*). All communications are supposed to be between 16-bit computers.

The information computer A wants to send to computer B is first transferred, under program control, to a fixed part of the memory of computer A, specially reserved for this purpose. We shall call it the SEND BUFFER. The length of this buffer is equal for all computers. When the transfer is completed, computer A puts a REQUEST WORD in the SEND REGISTER. At the same time, the SEND ADDRESS REGISTER (SAR) is set to its initial position (see par. 7). This initial position is the address of the first word of the SEND BUFFER. The REQUEST WORD is composed as follows:

MSB	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	LSB
	LENGTH OF MESSAGE (8b)								ADDRESS (5b)					PRIORITY (3b)			

The PRIORITY is from 1 to 7, with 7 the highest priority. The ADDRESS part indicates the computer B. With 5 bits, 32 different computers can be addressed. The LENGTH OF MESSAGE part is the complement of the number of words to be transmitted. This number cannot be higher than allowed by the length of the SEND BUFFER. When the SEND BUFFER is longer than 256 words, the LENGTH OF MESSAGE part consists of the 8 most significant bits of the complement of the number of words to be transmitted. This LENGTH OF MESSAGE part is not absolutely necessary: one could simply transmit the whole SEND BUFFER each time a communication is made. It is included here because it can speed up the communication when a large number of short messages is exchanged between the computers.

Via the line drivers, the REQUEST WORD is transmitted to the link. There, the REQUEST WORD is processed and, if a dataway is free and computer B not occupied with another communication, a connection is made between computers A and B. We will come back later to the hardware of the link. The DATA lines and lines TA, TB, SER and RER of both computers are then interconnected and the lines SEND ENABLE and RECEIVE ENABLE* are brought to logic 1 to indicate that the connection is made. The FLAG* of computer B is set to 1 and the RECEIVE ADDRESS REGISTER* (RAR) is set to its initial position. This initial position is the address of the first word of the RECEIVE BUFFER*, a reserved part of the memory of computer B, equal in length to the SEND BUFFER of computer A. The LENGTH OF MESSAGE part of the REQUEST WORD is put into the 8 most significant bits of the register WORD COUNT*.

Now everything is ready for the transfer of the message from computer A to computer B, under control of the CONTROL UNITS of both interfaces. Transfer to and from the computers is by the cycle stealing method via the direct memory access of the computers.

Interface B begins by sending a TA* pulse. When this pulse is received by interface A, the first word of the MESSAGE, as indicated by the SAR is put into the SEND REGISTER. The SAR is incremented by 1 and the word is transmitted to the RECEIVE REGISTER*, together with a pulse TB. When this pulse is received on the TB* line, the word is read into the first memory place of the RECEIVE REGISTER*, as indicated by the RAR*. The RAR* and the WORD COUNT* are incremented by 1 and a second pulse TA* is transmitted to interface A, in order to initiate the transfer of the second word of the message. This continues until an overflow is detected on the WORD COUNT*. This means that the message is complete and a pulse is transmitted on the END* line. This pulse is detected by the link hardware; the connection is broken and SE and RE* are brought back to 0. When SE is 0, the CONTROL UNIT sends an interrupt, to inform computer A that the transfer is completed. At the same time, an interrupt informs computer B that there is a message in the RECEIVE BUFFER*. Under program control, this message is transferred to another section of the memory, so that the RECEIVE BUFFER* is ready to receive new information. The FLAG* is put back to 0, to indicate that computer B is no longer occupied. We will come back later to the nature of the message and the means of handling it.

For certain processes, for instance a DTS transfer, the DMA may not be accessible for a certain time. The computer has to signal this by sending an inhibit to its interface. This inhibit blocks the transfer to or from the BUFFERS and also the TA or TB pulses. This stops the communication. When the inhibit disappears, the communication can resume at the point where it stopped.

4. THE LINK

The general configuration of the link is shown on figure 1. The RECEIVE UNIT (RU) and SEND UNIT (SU) are shown on figures 3 and 4 respectively. All RU's are mutually connected by bus-lines and have, moreover, one individual line to the RECEIVE CONTROL (RC). In the same way, all SU's are mutually connected by bus-lines and have also each one individual line to the SEND CONTROL (SC). The bus-lines on the send and receive sides are joined via line drivers in the link controllers RC and SC. Figure 5 illustrates the procedure by which the link controllers establish a connection.

The connections to the bus-lines are in the "wired or" configuration. This supposes negative logic with TTL (+3V for logic 0 and +0.2V for logic 1). Any number of dataways is possible but only one is drawn in detail, for clarity.

To fix the ideas, we suppose a REQUEST WORD is present on the DATA lines of the RU. These DATA lines are not yet connected to the dataway, so FF2 is "reset" and we further suppose FF1 to be "set". The PRIORITY part of the REQUEST WORD (D0 to D2) is now connected to the DECODER. This DECODER is somewhat special: when the binary input is 5, for instance, the output lines 1 to 5 are logic 1 and the lines 6 and 7 are logic 0. The output lines of all DECODERS are connected to the bus-lines P1 to P7 in the "wired or" mode. Only for those RU's with the highest priority present, the output S of the COMPARATOR is logic 1.

The RU's are scanned sequentially by the RC, by putting a 1 on the individual N lines. For those RU's with the highest priority present, the Q RESPONSE is 1 and the scanning stops. At the same time, the ADDRESS part of the REQUEST WORD appears on the bus-lines A0 to A4. This ADDRESS part is decoded in the SC, which sets to 1 the individual N line of the SU corresponding to the ADDRESS. When the corresponding computer is free (FLAG is 0), the B RESPONSE is 0. Now the RC and the SC generate FF2 ENABLE, FF3 ENABLE and STROBE. This "sets" FF2 and FF3 and establishes the connection between the computers. SEND ENABLE (SE) and RECEIVE ENABLE (RE) become 1 and the data lines D0 to D2 are no longer connected to the PRIORITY lines P1 to P7. The Q RESPONSE becomes 0 and the scanning of the individual N lines by the RC resumes.

Now, suppose that during the scanning a Q RESPONSE of 1 is detected but the computer requested on the address lines is not free (B RESPONSE of 1). Then FF1 DISABLE is generated by the RC and this particular REQUEST WORD no longer competes for priority. The scanning can continue for lesser priorities. When any END signal is detected, the situation is changed. Now FF1 ENABLE is generated by the RC and all REQUEST WORDS can compete again.

When an END pulse is detected by the SC and RC, the pulses STROBE and BREAK are generated on that dataway on which the END pulse was detected. FF2 and FF3 are reset and the connection is broken.

5. THE MESSAGE

The MESSAGE is the information that is transmitted from the SEND BUFFER of one computer to the RECEIVE BUFFER of another computer. For ease of communication, its form should be standard. The MESSAGE is composed of a HEADING and the RECORD. The HEADING can be, for instance, 3 words long and can contain the following information:

- number of words in the message
- sending computer
- receiving computer (for safety)
- identification of the subroutine for reading the RECORD
- possibly other information.

The RECORD can have any length within the limit of the SEND BUFFER. Usually it will be in one of the following two forms:

a) a communication in an intermediate-level language, to be interpreted by the receiving computer,

b) program sections or tables to be put in the core of the receiving computer without alteration.

A long STORE occupies valuable memory space and also occupies the dataway for a long time. During this time, a message with a higher priority cannot get access to this dataway. A BUFFER of 256 words is probably long enough for communications of type a). Communications of type b) are often longer but a simple sub-routine can cut them into sections which are transmitted sequentially.

6. ERROR DETECTION AND TIMING

The problem of timing and error detection is treated by E. Asseo (CERN/MPS/CO 71-11). The DATA words will only be accepted when the word on the lines has not changed for, say, one microsecond. This change is detected by the COMPARATOR (see figure 2). The same is done for the timing pulses TA and TB. For details, see the report mentioned above. If a bad transmission is detected at either side, this is communicated to the other side by means of pulses on lines SER or RER (see figure 2). The communication is broken and the sending computer is warned, by interrupt, that it has to transfer the REQUEST WORD again to the SEND BUFFER.

The transfer of one word will take from 10 to 20 microseconds, depending on the distance. The scanning of the priorities will take about 2 microseconds per scanning step.

7. COMPUTER MEMORY PROTECTION

The first thing to assure is that the RECEIVE REGISTER cannot write outside the RECEIVE BUFFER. Suppose the RECEIVE BUFFER is 256 words long. Then, the 8 most significant bits of the ADDRESS REGISTER can be hard wired with semi-permanent straps. The 8 least significant bits form a counter. To set the ADDRESS REGISTER to its initial position means to set the 8 least significant bits to zero. Writing outside the RECEIVE BUFFER is now a physical impossibility.

The communications in intermediate language use symbolic addresses. The interpreter in the receiving computer can have a list of addresses of arrays, where writing is allowed. If writing outside these areas is attempted, the command will be ignored.

Another form of communication is the transfer of whole program sections or tables, to be put into the memory without alteration. The HEADING of such a message must identify the subroutine for writing the RECORD into the memory. In this way, writing is always under control of the receiving computer.

8. PRIORITIES

The messages must reach their destination within a certain time, otherwise their information is no longer relevant. Most data are refreshed regularly and the transmission delay must be smaller than the time between refreshings. The smallest transmission delay that can be guaranteed is of the order of 20 ms. In certain cases, buffer registers are necessary, which accumulate the data and transmit them at regular intervals.

In certain cases information must be transferred very quickly after it is obtained (for instance in feedback loops). In those cases, the connection can be requested a few ms before the information has to be transmitted. We will give those informations the highest priority: 7. The connection is established normally but the transfer from the SEND BUFFER is inhibited by the CONTROL UNIT. When the information is available in the SEND BUFFER, a new REQUEST WORD is read into the SEND REGISTER, with lower priority, so that the inhibition no longer operates and the data transfer proceeds in a normal way. This priority should be used sparingly, because the dataway is occupied longer than is necessary for a normal transfer.

It is logical to assign the other priorities also in function of the maximum transmission delays that are allowed. A possible way to do this is given below:

priority 7	dataway connection reserved beforehand
priority 6	20 ms max. delay allowed
priority 5	50 ms max. delay allowed
priority 4	200 ms max. delay allowed
priority 3	500 ms max. delay allowed
priority 2	(to be transmitted only during quiet
and 1	(periods of the cycle.

Another problem is to arrange the flow of messages to the SEND STORE. This flow is under program control and normally there is some definite sequence in which the communications should take place. At quiet moments there is no problem, because the messages leave the SEND BUFFER in the proper sequence. At moments of peak communication, however, there will be a pile-up of communication requests. An efficient way has to be found for handling these requests, so that the most urgent ones leave the computer first. The computer has of course the option to change the priorities for its messages according to the circumstances.

9. CONCLUSION

We have shown that it is possible, with relatively simple hardware, to make a versatile auto-controlled link, which can handle peak loads efficiently. It is doubtful whether a control computer, specially programmed for controlling the link, could do better. In fact, to do better, the control computer should be aware, at all times, of the time-table of all possible communications. It is difficult to program the computer in such a way in a changing environment. For special cases, like machine development, it is completely impossible.

The link is transparent to the user. The only thing the programmer has to know about the link is the structure of the REQUEST WORD, the priority code and the meaning of a few interrupts and a flag.

The link is modular and can easily be adapted for any number of computers. This is important for an evolving computer configuration. A possible final system structure and a way of evolving to it are proposed by H. Kugler and H. Riege (unpublished). The computer link, described in this note, is well adapted to such a configuration.

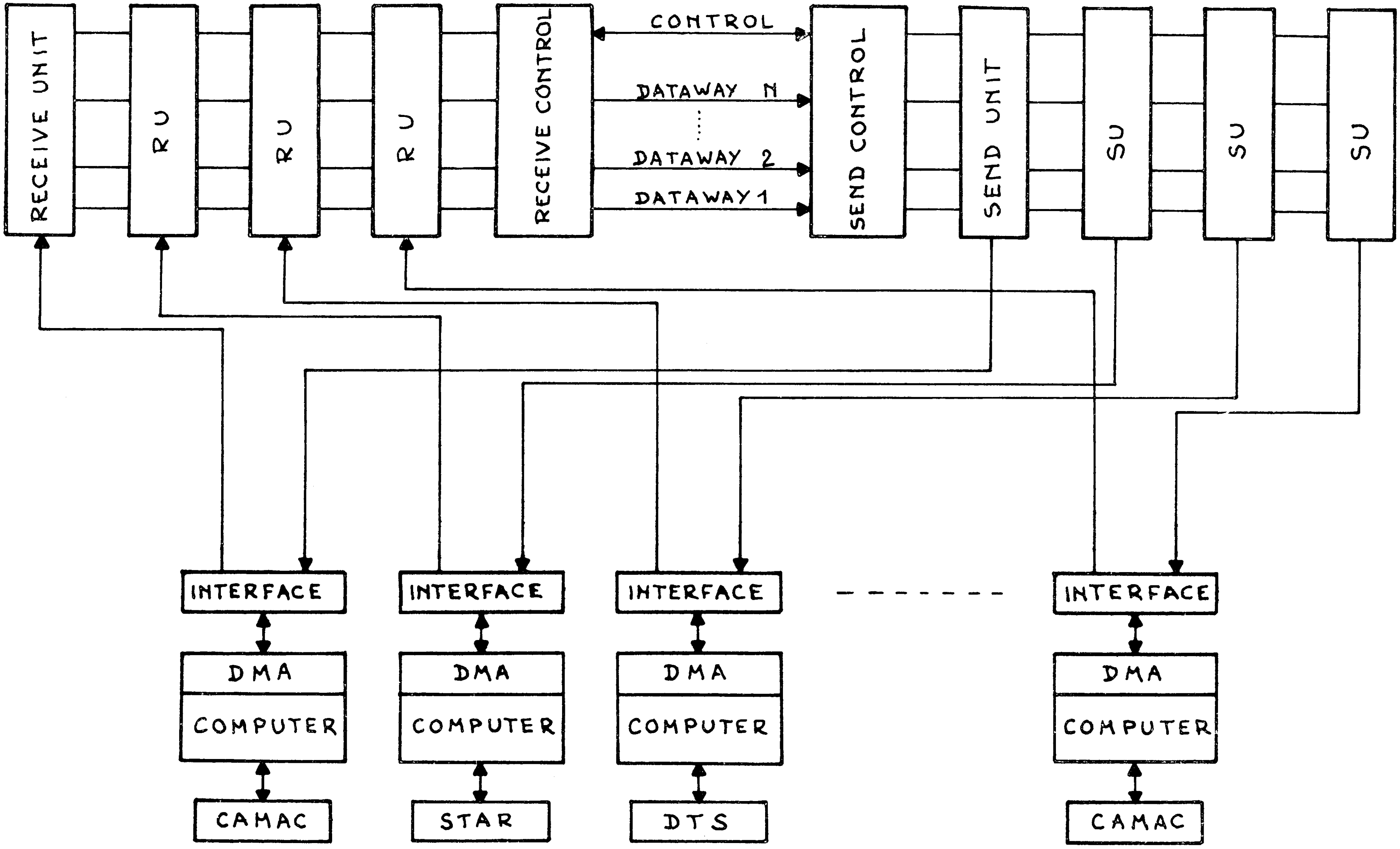

FIGURE 1 : GENERAL CONFIGURATION

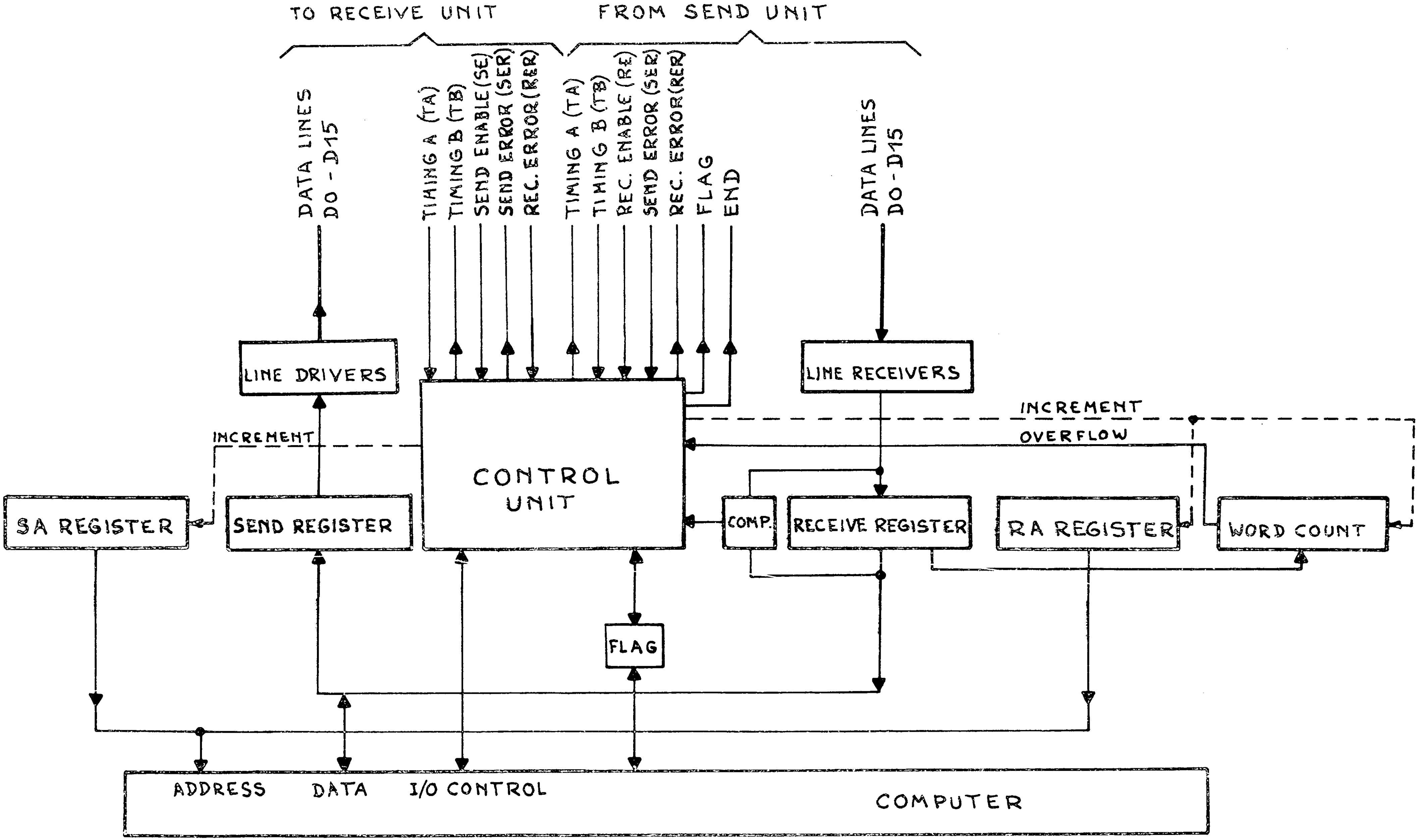

FIGURE 2 : THE INTERFACE UNIT

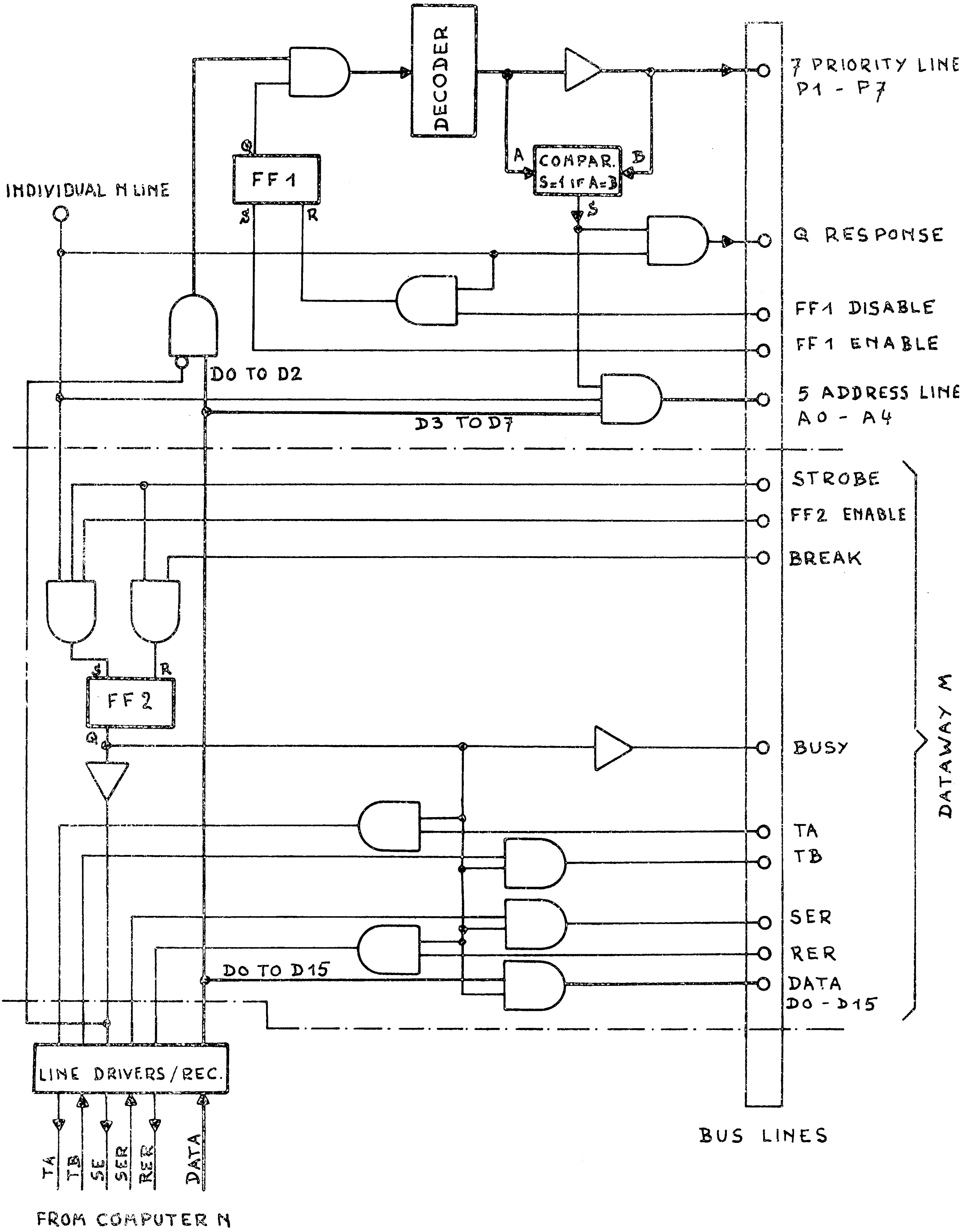

FIGURE 3 : RECEIVE UNIT

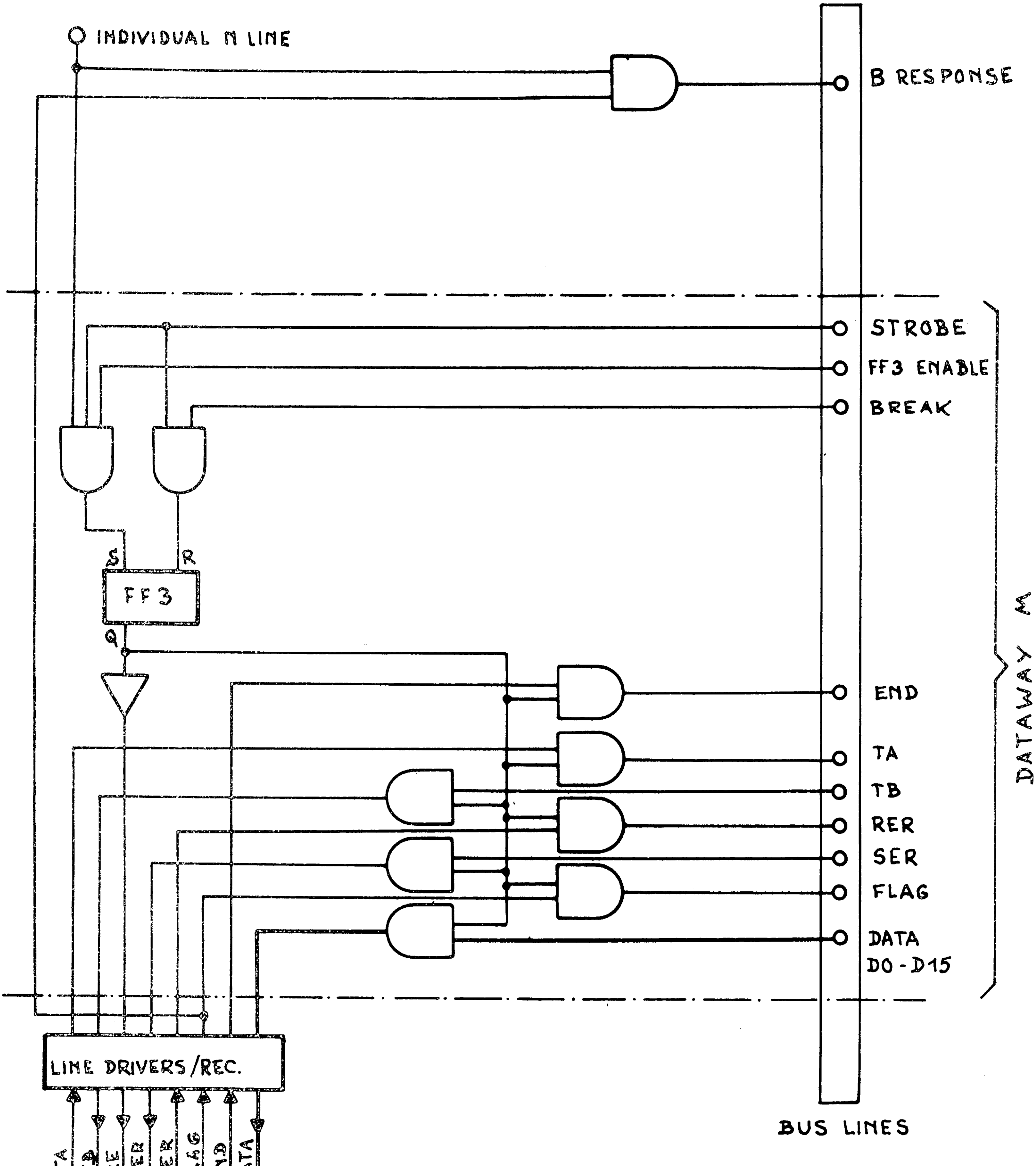

FIGURE 4 : SEND UNIT

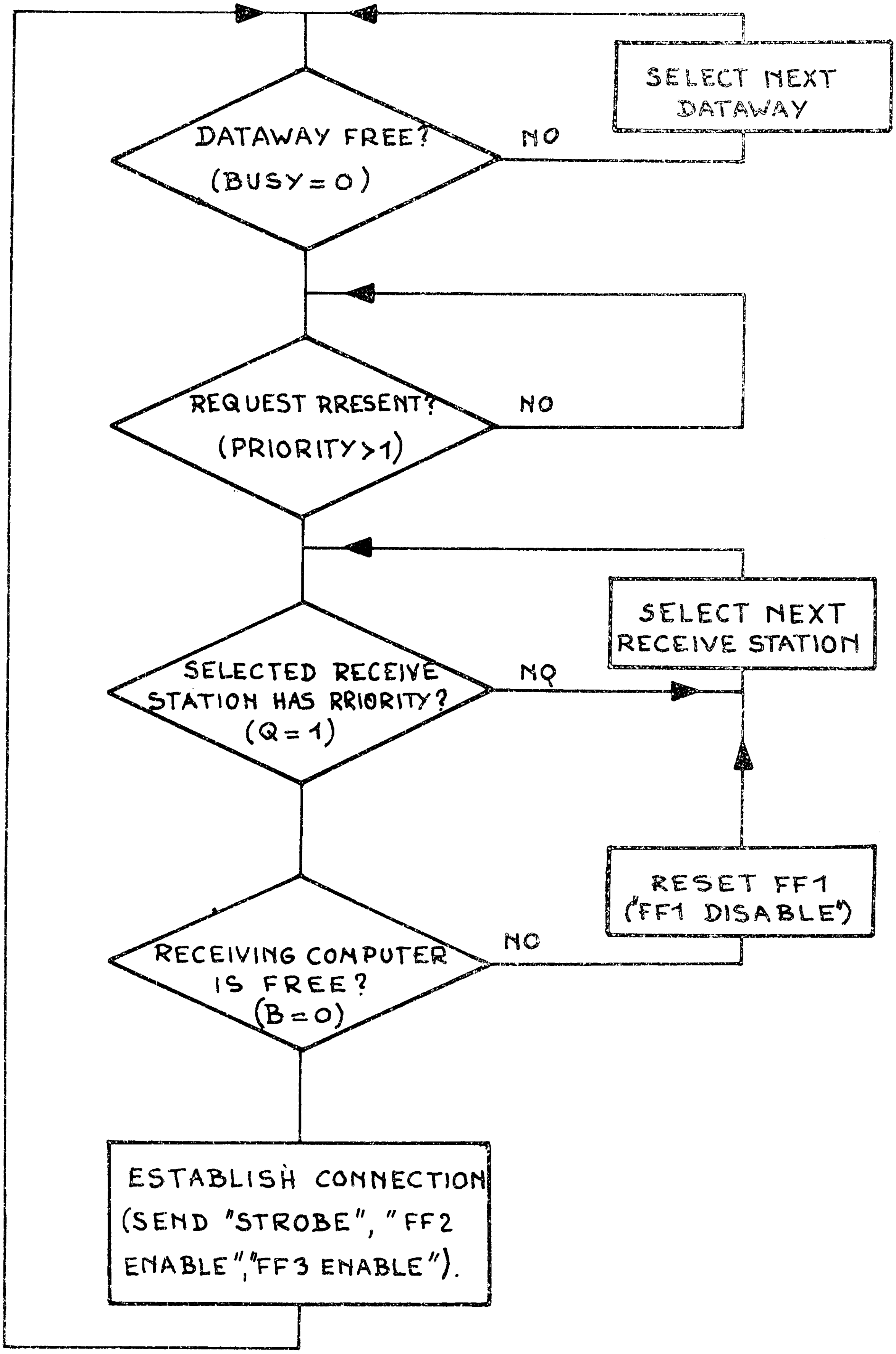
SELECT NEXT DATAWAY
DATAWAY FREE? (BUSY=0)
NO
REQUEST RRESENT? (PRIORITY>1)
NO
SELECT NEXT RECEIVE STATION
SELECTED RECEIVE STATION HAS RRIORITY? (Q=1)
NO
RESET FF1 ("FF1 DISABLE")
RECEIVING COMPUTER IS FREE? (B=0)
NO
ESTABLISH CONNECTION (SEND "STROBE", "FF2 ENABLE","FF3 ENABLE").

FIG.5: ORGANIGRAM FOR SEND- AND RECEIVE CONTROL UNITS