Automatic Control and Computer Sciences, Vol. 22, No. 4, pp 44-51 (1988)

STRUCTURAL ORGANIZATION AND INFORMATION INTERCHANGE PROTOCOLS FOR A FAULT-TOLERANT SELF-SYNCHRONOUS RING BASEBAND CHANNEL

V. I. Varshavskii, V. Ya. Volodarskii, V. B. Marakhovskii,
L. Ya. Rozenblyum, Yu. S. Tatarinov, and A. V. Yakovlev

Avtomatika i Vychislitel'naya Tekhnika,
Vol. 22, No. 4, pp. 48-55, 1988

UDC 681.327:519.713

The authors develop the concept of fault-tolerance in relation to the organization of a systems layer interface for a microcomputer LAN with a ring baseband channel. They investigate problems of design of the structure and information interchange protocols in the baseband with allowance for self-synchronous implementation of the interfacing hardware.

The ever more stringent reliability and performance requirements for local-area networks (LANs) based on multiprocessor and multicomputer architectures with a bus or ring baseband channel, cannot be fully satisfied by the traditional methods of enhancing reliability through the use of processor redundancy. In such systems, the baseband channel is a bottleneck which prevents achievement of the requisite level of reliability, since malfunctioning of any line of the baseband causes the entire system to fail. Paper [1] discussed aspects of circuit-engineering implementation of a self-synchronous fault-tolerant interface based on the common-bus principle, while paper [2] proposed methods of translating the formal specifications of physical layer protocols into self-synchronous interface adapter arrangements. In this paper, we will develop the concept of fault-tolerance in relation to organization of a systems layer interface for a microcomputer LAN with a ring baseband channel. On this basis, we have developed a scheme for a self-synchronous (aperiodic) network adapter, which executes, in addition to the necessary protocol functions, diagnostic, fault-localization, and self-repair algorithms.

Structural organization and requirements for baseband channel. Figure 1 shows a ring structure of a baseband channel for a microcomputer network. Each computer has a standard interface to the baseband, and operates in conformity with the standard (e.g., Q-bus standard). The organization of the interface from the baseband side involved the satisfaction of the following requirements: transmission speed in the baseband of at least 5 Mbit/sec; pairing of any two and detection of three line faults in the channel and associated hardware in each segment of the ring; disconnection of controller from baseband in response to a signal from the "native" or from a "foreign" computer, and self-disconnection of the controller in response to a fault signal from its transmit/receive part; arrangement for informing the computer that the controller is in a line-repair mode, after which the computer must perform a number of operations related to restoral of baseband operational status; bidding for channel performed by controller on the basis of message priorities, using token-access mechanism.

The network layer protocol (i.e., higher than the one under consideration) is implemented in software in the computers, which exchange messages via their controllers. Each message consists of a header, message body, and trailer or terminator. The header contains information on the message priority and type, the addresses of the recipients, the address of the sender, and the message length. The trailer contains the checksum of the message. The required modes of interchange between computers are one-to-one, one-to-all, and one-to-many. Message transmission provides for the fact that the act of reception (transmission) by the controller at the local Q-bus interface and the act of transmission (reception) in the ring channel must be "decoupled" in time, so that the controller performs message buffering at both the channel input (in the transmit mode) and channel output (in the receiving modules).

To ensure correct transmission over the channel wires, regardless of the "skew" in their delays, a self-synchronizing code (SSC) is employed [3]. The chosen SSC was an

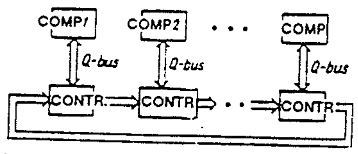

Fig. 1. Ring structure of baseband channel.

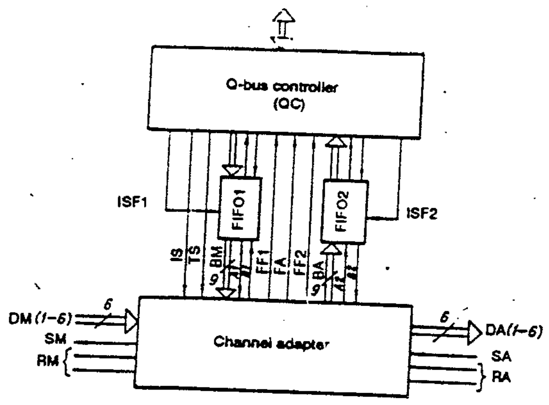

Fig. 2. Consolidated structural diagram of controller.

optimal equilibrium code (OEC) of type C_6^3, which has 20 working sets and permits transmission of one half-byte of information during one interaction of adjacent controllers. Half-byte transmission makes it possible to satisfy the requirement of economy in the number of connecting wires of the channel, and also the requirement that the speed be not less than 5 Mbit/sec (the permissible delay in transmission of one half-byte is equal to 800 nsec, which is equivalent to successive actuation of 150-300 gates in a VLSI circuit). To transmit a half-byte, we adopted a self-synchronous C_6^3 OEC code and a transmission protocol with a spacer of "all zero's" type [1].

Figure 2 shows a consolidated structural diagram of the controller. It contains four units which are assumed to be implemented in the form of self-contained LSI circuits: a standard Q-bus controller (QC); a transmission buffer (FIFO1); a reception buffer (FIFO2); and a self-synchronous channel adapter (CA). In the context under consideration, the QC is of no particular interest, and can be implemented in accordance with the conventions provided for in the Q-bus standard. The FIFO1 and FIFO2 circuits are self-synchronous buffer storage circuits of first-in first-out type. An example of such a storage arrangement is described in [4]; an experimental LSI version was developed.

Our central object of concern is the CA. We will describe the lines linking the CA and the QC and both FIFO. Over the internal (for the controller) nine-bit buses BM and BA a byte of code is transmitted, this being synchronized by the maintenance signals A1 and A2 and acknowledgment signals B1 and B2 (the phenomenon of "skewing" on this interface can be disregarded). The IS ("initial set") signal is generated in each of the controllers in response to a signal from its "native" computer at the instant that the entire system is reset to the initial state. The TS ("token set") signal is generated in the module which is the system initiator ("master"), with the result that the module in question initiates the first channel-bidding cycle by generating an initial token. In the event that an error arises in transmitting the status of the master from the systems initiator to one of the channel modules, the system is reinitialized with the necessary self-recovery, after which the function of master is passed to another initiator, "ASSISTANT," in which a TS signal is also generated. The ISF1 and ISF2 signals are issued from the QC and indicate that the information in FIFO1 and FIFO2 should be reset. They are generated prior to commencement of operation and after detection of a fault in FIFO1 and FIFO2, when it is necessary to eliminate the remainder of an untransmitted message for subsequent repeat transmission, which is effected on the network software layer. The signals FF1 and FF2 are indicates of FIFO1 and FIFO2 faults respectively; they are generated by the CA after establishment of a fault in the operation of the corresponding FIFO when data is "pumped" in or out. In response to FF1, the CA must perform an initial-set of FIFO1 and organize repeat transmission, while in response to FF2 the CA must reset FIFO2 and organize transmission of an overhead message to the source of the unreceived message, to the effect that the latter must be retransmitted. However, these issues must be resolved specifically on the higher network layer of intercomputer interchange. By means of the FA ("fault") signal, the CA informs its own computer (via the QC) that there is a fault in the channel, thus providing a warning that repeat message transmission and reception is necessary. The DM and DA buses are the reception and transmission buses for a half-byte of data from the channel to the CA and from the CA to the channel respectively. Data are transmitted over these buses in half-bytes in C_6^3 OEC code. Despite the one-way transmission in the ring, the buses are bidirectional, since they can handle signals in both directions in the fault-localization and self-repair mode. The acknowledgment-signal lines SM and SA are also bidirectional for this same reason. In accordance with the requirements for avoiding two line faults, the channel buses include two spare or reserve lines (RM and RA) for use as information or acknowledgment lines. The back-up principle is a sliding one [1], i.e., when the i-th line fails there is a switching of all lines beginning with the i-th with number i + 1 to i, such that the physical acknowledgment line becomes an information line, and a spare line is connected in its place. It should be pointed out that, in addition to the wire itself, the line hardware includes the backbone transmitters and receivers that operate on this line.

Purpose and functions of interaction layer under consideration. In accordance with the generally accepted terminology and classification [5], the protocol layer under consideration is the physical layer of a local-area network with a ring baseband channel. The primary service function of this layer with respect to the next higher interaction layer is reliable transfer of the sequence of message bytes coming from computer 1 (transmitting computer) to the QC and buffered in FIFO1, to one (or more) computer 2 (receiving computer) via its own QC with buffering in FIFO 2. This structuring of the layer assumes that the ring baseband channel proper consists of a chain of CA that are capable of receiving data from FIFO 1 (in bytes) and from the channel (in half-bytes) and of transmitting it to FIFO2 (in bytes) and to the channel (in half-bytes).

To transfer data to the channel, the CA must bid for the channel, using the token access method with priorities. After bidding for the channel, the CA that has seized it acquires "master" status and can initiate transfer of a sequence of half-bytes to its output port. Transmission of this sequence around the ring is organized as a asynchronous pipeline process, such that each half-byte advances to the next successive CA when a free place is available in it.

Thus, the physical layer of the LAN under consideration must handle the following task: Transmit the byte sequence of the "master" message in its FIFO1 to all the adresses in this message, and enter this message byte by byte into their FIFO2. To achieve this aim, it is necessary to specify precisely the structure of the message arriving from the FIFO 1 of the master, and of the message transferred to the FIFO2 of the recipient. The description is given in the customary BNF notation.

1. The message in FIFO1 has the following structure:

```
⟨message 1⟩ ::=⟨header 1⟩ [⟨information part⟩] ⟨terminator 1⟩,
⟨header 1⟩ ::=⟨byte of priority and number k of recipient address bytes⟩;
⟨bit 9=∅⟩ {⟨recipient address byte⟩; ⟨bit 9=∅⟩}^k,
⟨information part⟩ ::={⟨information byte⟩; ⟨bit 9=∅⟩}^n,
⟨terminator 1⟩ ::=⟨end-word byte⟩; ⟨bit 9=1⟩.
```

2. A message in FIFO2 has the following structure:

```
⟨message 2⟩ ::=⟨header 2⟩ [⟨information part⟩] ⟨terminator 2⟩,
⟨header 2⟩ ::=⟨state byte⟩; ⟨bit 9=∅⟩,
⟨terminator 2⟩ ::=⟨end-word byte⟩; ⟨bit 9=1⟩.
```

We should note that the addressing of the recipients in the address bytes is done by means of a unitary code. It should be stipulated that the other structural elements, enumerated in the requirements, for a message transferred from one computer to another may be ignored by the given interchange layer, since they are transferred as ordinary information bytes. In particular, since the end of the message is "tagged" by a byte for which the value of tag bit 9 is equal to 1, in order to determine the end of the message in the CA it is not necessary to talley the number of bytes coming from FIFO1, and therefore the "envelope" of this layer does not include either the message length or type. It is also understood that the checksum is not singled out on this layer, since transmission is by means of SSC (it is contained at the end of the information part of the message), which is stable to one-way distortions, and checking of accuracy by means of the checksum can be passed on to the next successive layer.

Channel bidding protocol. The process of bidding for the channel for the purpose of subsequent message transfer is organized on the basis of the token access method, using a dynamic priority mechanism.

At the outset of operation of the system (before initial set), one of the modules (called the "main master") generates the TS signal, and then this module issues the token half-byte T1 to the channel (tokens are encoded by free C_6^3 OEC codes). In subsequent operation of the system, the source of the initial token T1 is the current master, which has just completed delivering the end byte of its message.

Subsequently, each module, upon receiving T1 at its input from the channel, checks to see if there is a request-to-transmit signal from its FIFO1. (The sequence of these checks in the modules of the ring in effect comprises an arbitering-allocation procedure for priority requests of modules that are bidding for the channel.) If there is no such request, the module stays in the initial state and delivers T1 to its output to the channel. If, at the instant of arrival of T1, the module has a request, it acquires the status of a "bidder" and delivers to the output first the token half-byte T2, and then its priority half-byte N_c, obtained from FIFO1 in the first message byte.

If a module in the initial state receives token T2 at its input, and then a half-byte with the current maximum priority number (N), it becomes an "observer" (if no request is present) and translates T2 and N from its input to the output to the channel, or (if a request is present) it becomes a "bidder," issuing T2 and then a priority number N_m that is equal to the greater of the following two: its own priority number N_c or the priority number N received from the channel. If a module that is in the observer state receives T2 and N at its input, it remains an observer and delivers T2 and N to the channel.

If a "bidder" module obtains token T2 and N from the channel, it delivers T2 and N to the output to the channel; if this number N is greater than its own number N_c, it issues token T3, if its number N_c is not less than the received number N, and goes into the "master" state, after which it initiates message transfer from its FIFO1.

A module in the observer status passes token T3 received at the input without alteration to the output, and acquires "recipient-observer" status, activating address reception from the channel. A module that has bidder status, upon receiving T3, becomes a "recipient-bidder" and allows T3 to pass to the output to the channel, after which it activates address reception from the channel. A module that has become a master, upon receiving T3 from the channel, also becomes a "potential recipient" ("by compatibility," as it were, since it involves two subautomata operating in parallel, namely the subautomata that handle transmission and reception respectively), and activates address reception from the channel.

Thus, the bidding process is completed, and all modules go into one of the following states: "master with recipient," "recipient-bidder," "recipient-observer."

Let us clarify the purpose of differentiating the recipient status. The "recipient-bidder" status is introduced in order to differentiate a recipient who wishes to bid for the channel himself but cannot do so in the current conflict-resolution process, since his priority is insufficiently high. To avoid the "death-by-starvation" effect, the proposed protocol provides for a mechanism of dynamic priority incrementation relative to the priprity that is initially specified in the message, i.e., after a master has completed transmission of a message, all modules with recipient-bidder status increase their priority number by 1/2. This ensures that the resources of the baseband are equitably allocated among all the modules.

The "master with recipient" status derives from the fact that a master, in transmitting a message in a "one-to-all" or "one-to-many" mode, may also address itself, e.g., to check the message and by whom and how it was received. Therefore its receiving FIFO2 may also have the module's own message in it. Since the message contains an end word, it is used to supply information on the recipients who correctly received the given message.

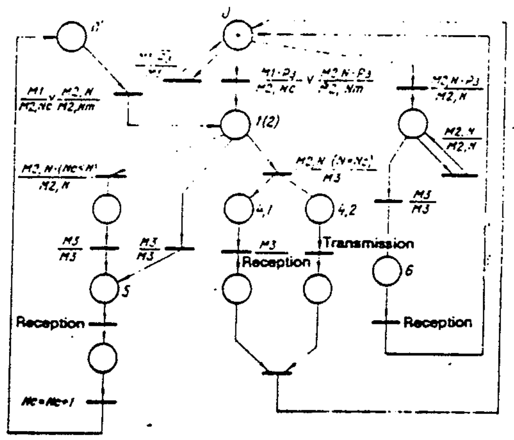

Fig. 3. Marked Petri net that specifies channel-bidding protocol.

Figure 3 shows a marked Petri net (MPN) with "viable" and "safe" marking, that comprises the formal specification of the channel-bidding protocol. The position numbers corerspond to the states that the module can assume: 0, initial state; 0', initial recipient-bidder state; 1 (2), bidder; 3, "observer"; 4 (4.1 and 4.2), "master with recipient"; 5, "recipient-bidder"; 6, "recipient-observer." The user of Petri nets as opposed to algorithm flowcharts enables us to adequately express the parallelism in the master upon transmission and reception of its message. Unmarked positions are necessary additional positions (that do not carry a semantic load). Transitions of the network are indicated as follows: the expression above the bar gives the value of the input from the channel and internal predicates of type RZ (RZ = 1 if the condition "request present" is true), $N = N_c$ and $N_c < N$; the expression below the bar gives the value of the output to the channel. In addition, one of the transitions corresponds to the operation of incrementing the priority N_c.

Addressing principle for modules in channel. It was pointed out in [1] that, for systems with a backbone architecture, it would seem that a radical solution of the problem of collective acknowledgment in organizing "one-to-all" interchange does not exist, if we require that the operation of the interface be entirely independent of the delays of the interchange participants and the connecting lines. For systems with a ring architecture, however, the principle of self-synchronization can be fully implemented. Here a message delivered by the master to the ring and received by him at his input can function as a collective acknowledgment.

Let us consider the addressing principle for modules in a ring channel. Assume that a unitary code is employed for addressing. Then the length of the address field of the message is equal to the number of modules in the system. A bit number of 1 in the address field indicates the number of the addressed module to which the message is being sent. Thus, the presence of some (all) 1's in the address field specifies a "one-to-many" ("one-to-all") mode of interchange. In addition, to ensure that the address-decoding mechanism is self-checking, each of the modules must ensure a one-to-one correspondence between the physical ("geographical") position of the module relative to the master and the number of the corresponding bit in the address field. This is achieved as follows.

Let us consider some module that is in position i from the master in the direction of message transfer. The physical position of the module relative to the master corresponds to the presence of a 1 in the i-th bit of the address (reckoning from the high-order bit). Upon transmission around the ring, each module records the high-order address bit, and then transmits the address field to the ring with a one-bit shift in the direction of the high-order bit. Thus, the i-th module receives an address field whose high-order bit contains the "choice" or "nonchoice" marker of the i-th module. This marker is stored until the next module-addressing event. If it is equal to 1, the CA transfers the message to FIFO2 and translates it to the output channel. If the marker is 0, only translation in the direction of transmission is performed. This organization of addressing eliminates hardware decoding of the address, thus removing the problem of decoder self-testability. It is necessary, however, that the address field store the

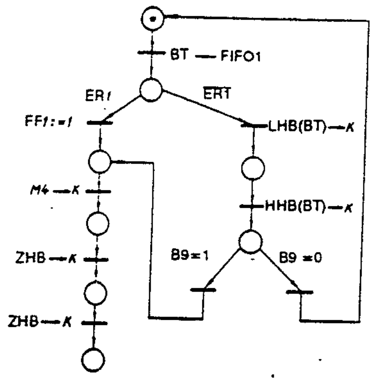

Fig. 4

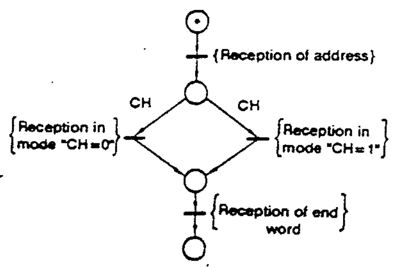

Fig. 5

Fig. 4. Message transmission protocol.

Fig. 5. Consolidated representation of message-reception protocol.

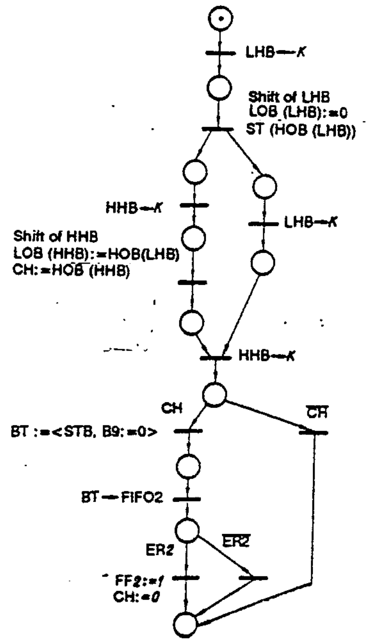

Fig. 6. Protocol for reception and recognition of recipient address.

LHB low order half byte
HOB - high order bit
LOB - low order bit

relative addresses of the selected modules. This imposes additional functions on the higher interaction layer. As a possible implementational version of addressing on the higher layer, we can propose that each computer store a table of correspondence of relative and absolute addresses.

In the process of operation, the system may be reconfigured because of malfunctioning of module i in the r-th geographical position, such that the former (r + 1)-th module becomes the r-th, and so forth. To reconfigure the table of correspondence, in accordance with the higher-layer protocol, a procedure must be carried out in which the master exchanges a message of "Who are you?" type with each module (successively generating a 1 in each of the address bits), thus learning the absolute addresses of its newly reconfigured relative neighbors. Upon receiving a 1 in the r-th position, the r-th module in order issues a "This is who I am" message in a "one-to-all" mode (without knowing who sent the request), and this message is received by the master, who records the new absolute address of the module in the r-th position of the table. After the table is reconfigured, the master can transmit its own table to the remaining module. These modules align their tables with allowance for the shift of table positions relative to the master.

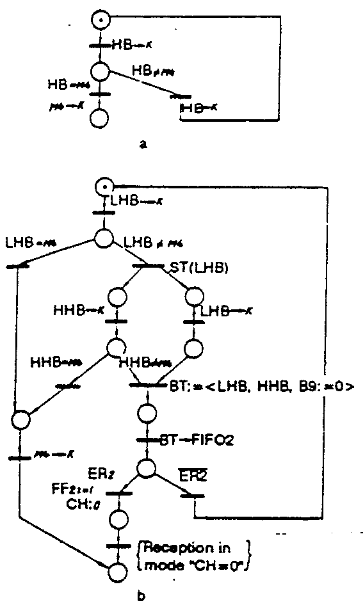

Fig. 7

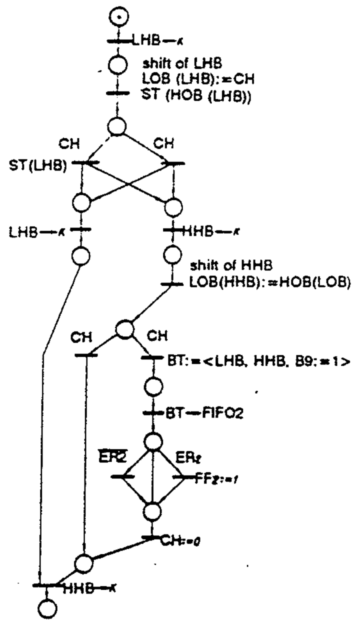

Fig. 8

Fig. 7. Reception protocols for information part of mesage in "unselected-recipient" (a) and "selected-recipient" (b) modes.

Fig. 8. Recpetion protocol for end word.

Brief description of message transmission and reception protocols. Message transmission by one of the modules that has gained "master with recipient" status begins immediately after this module issues token T3. Then the master issues message half-bytes in a pipeline mode, beginning with the address and completing the message with an end marker T4, which is then followed by a pair of null half-bytes which serve to "assemble" the end word over the entire channel. Figure 4 shows an MPN that specifies the transfer protocol that is implemented by the transmitting part of the master. The following notation is adopted: BT, byte; ER1, error marker upon reception of byte from FIFO1; FF1, fault signal for FIFO1; LHB, low-order half-byte; HHB, high-order half-byte; C, channel; B9, bit 9 in BT; ZHB, half-byte consisting of zero's. Message reception (Fig. 5) involves address reception, fixing of the choice-of-recipient marker CH, and execution of the reception protocol for the information part in a chosen-recipient (CH = 1) or unchosen-recipient (CH = 0) mode; reception terminates with the end-word reception procedure. Figure 6 shows the address reception and recognition protocol. Figure 7 shows MPN of the reception protocols in modes CH = 0 (a) and 1 (b). The end-word reception protocol, shown in Fig. 8, is analogous to address reception in many respects. These figures employ the following additional notation: HB, half-byte; STB, state byte; LOB, low-order bit; HOB, high-order bit; ST(X), "store value of X"; FF2, fault signal for FIFO2; ER2, error marker in transmitting byte to FIFO2. It should be pointed out that the receiving part of the master executes the reception protocol in the customary fashion, but does not deliver message half-bytes to its own output channel.

On the basis of the above specifications, we formally constructed, in the form of MPN, a logical control diagram for adapter operation, which requires an independent analysis.

REFERENCES

1. V. I. Varshavskii, V. B. Marakhovskii, L. Ya. Rozenblyum, Yu. S. Tatarinov, and A. V. Yakovlev, "Approach to reliable circuit-engineering implementations of physical layer protocols of network architecture," AVT [Automatic Control and Computer Sciences], no. 6, pp. 76-81, 1986.

2. V. I. Varshavskii, V. B. Marakhovskii, L. Ya. Rozenblyum, Yu. S. Tatarinov, and A.V.Yakovlev, "From physical layer protocol specification to hardware implementation," AVT [Automatic Control and Computer Sciences], no. 5, pp. 82-88, 1987.

3. V. I. Varshavskii (Editor), Automation Control of Asynchronous Processes in Computers and Digital Systems [in Russian], Nauka, Moscow, 1986.

4. V. I. Varshavskii et al., "Self-synchronous buffer storage," VINITI deposition number 6337-B87, 27 August 1987.

5. E. A. Yakubaitis, Local-Area Networks and Their Architectures [in Russian], Zinatne, Riga, 1985 [English translation by Allerton Press, 1986].

12 February 1988