(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 346 441 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
17.04.1996 Bulletin 1996/16

(21) Application number: **89901198.5**

(22) Date of filing: **16.12.1988**

(51) Int Cl.[6]: **G05B 19/05**, G06F 17/60

(86) International application number:
PCT/US88/04512

(87) International publication number:
WO 89/06010 (29.06.1989 Gazette 1989/14)

(54) PROCESS AND DEVICE FOR COMMUNICATING REAL TIME DATA BETWEEN A PROGRAMMABLE LOGIC CONTROLLER AND A PROGRAM OPERATING IN A CENTRAL CONTROLLER

VERFAHREN UND VORRICHTUNG ZUM ÜBERTRAGEN VON ECHTZEITDATEN ZWISCHEN EINEM PROGRAMMIERBAREN LOGIKSTEUERGERÄT UND EINEM PROGRAMM IN EINEM ZENTRALEN STEUERGERÄT

PROCEDE ET DISPOSITIF DE COMMUNICATION DE DONNEES EN TEMPS REEL ENTRE UNE UNITE LOGIQUE PROGRAMMABLE DE COMMANDE ET UN PROGRAMME DANS UNE UNITE CENTRALE DE COMMANDE

(84) Designated Contracting States:
DE FR GB IT

(30) Priority: **17.12.1987 US 134353**

(43) Date of publication of application:
20.12.1989 Bulletin 1989/51

(73) Proprietor: **AEG Schneider Automation, Inc.**
North Andover, Massachusetts 01845 (US)

(72) Inventor: **ROSEMAN, Brooks, Todd**
Asheville, NC 28806 (US)

(74) Representative: **Baillie, Iain Cameron et al**
c/o Ladas & Parry
Altheimer Eck 2
D-80331 München (DE)

(56) References cited:
EP-A- 0 136 485
EP-A- 0 174 231
EP-A- 0 200 365
US-A- 4 648 062
US-A- 4 704 704
US-A- 4 751 635
US-A- 4 791 556

- **Lotus Development Corporation, 1986, 1987 (Cambridge, Massachusetts), "Lotus Measure, Data Direct to Lotus Spreadsheets", see entire document.**
- **Lotus Development Corporation (Cambridge, Massachusetts), "Lotus Developer Tools, Comprehensive Developer Tools for 1-2-3 and Symphony", see entire document.**
- **Strawberry Tree Computers, Inc., August 1987 (Sunnyvale, California), "Analog Connection PC Software for the IBM PC", see pages 36-37, "Labtech Notebook for the IBM PC", see pages 38-39, "Labtech Real Time Access for the IBM PC", see page 40, and "Labtech Acquire for the IBM PC", see pages 40-41.**
- **Control Engineering 34 (1987) March, No. 3, Newton, MA, USA, pages 92-94, Michael Babb: "Implementing Distributed Control at the Loop Level"**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 346 441 B1

Description

Background of the Invention

This invention generally relates to programmable logic controllers that direct the automatic operation of such as machine tools, and process equipment to manufacture goods and chemicals, and particularly relates to furnishing true real-time control of such programmable logic controllers (PLCs) through a general purpose spreadsheet program operating in a personal computer.

Machine tools such as screw machines have come to be operated automatically. This eliminates the cost of manual labor in operating the machine and can achieve more uniform products. The automation originally occurred by arranging relays to perform the logical functions of proceeding through a manufacturing process. Switches sensed the movement of tools relative to the work piece and caused the opening and closing of the relays; the relays in turn caused the sequential selection and operation of different tools on the work piece.

Much skill is required however to design the system, to set-up the limit switches at the proper locations, to connect the switches to the proper relay solenoids and to connect the relay contacts to the proper motors, timers and solenoid controlled valves in achieving the automated operation. Changing even one dimension of a part being made or one process variable in a chemical reaction moreover required skilled labor to re-wire the control switches or relays. Such systems worked well in producing large numbers of parts from a single set-up, but the development of manufacturing fewer numbers of more customized parts on one machine made the frequent changes of the relay controlled machines too expensive.

Responsive to this rise in cost, PLCs replaced the relay controls. PLCs furnished a fixed wiring from the switches, motors, and solenoid valves to a general purpose logic device. This eliminated rewiring the controls for each different part or variation of a part to be made. The set-up for each different part or variation of a part then only required changing the logical commands in the logic device. With the advent of the microcomputer as the logic device, changing the logical commands in the logic device became a matter of re-programming the microcomputer. Skilled labor then only had to write one program to make each different part or variation of a part, and an unskilled operator then could load the different programs into the PLC as desired to make a desired part. This substantially eliminated costly set-up for each different part; the manufacture of that part or the running of that process was fixed, however, by the fixed nature of the program.

The fixed nature of the program does not always allow for varying automatic operation of the screw machine, for example as the material of the work piece varies, or for example as the characteristics of a process feedstock vary. Moreover, the fixed program does not allow an operator to vary the manufacturing or processing parameters as changes occur in the manufacturing or processing operation to achieve desired efficiencies and levels of quality.

The PLCs have been linked together in networks over co-axial cable to co-ordinate the operation of an entire manufacturing or processing plant. The network requires separate interface modules terminating each drop of the co-axial cable to perform the necessary handshaking for positive message transfer between the addressable registers of the PLCs. These messages usually comprise status information indicating the condition of the sensors and drives of each machine tool or processing station and control information commanding the operation of the drives or valves of the tools or stations. The messages also can contain numerical information indicating the number of parts made or the temperature or pressure of a point in a process. An example of such a PLC network can be seen in document EP-A-200365.

A centrally located programming device can also be connected to the network to change the program or sequence of logical steps to be implemented by any selected PLC on the network. Such a programming device can be a personal computer, for example an IBM or IBM-compatible personal computer, operating a program that displays relay ladder diagram charts that a skilled user develops to create desired logical steps for a PLC. The personal computer also contains a network interface circuit card, connected to the co-axial cable, that handles the network handshaking and positive message transfer in response to commands by the personal computer written to and from accessible registers on the circuit card. The registers of the interface card are available to programs running in the personal computer able to access them to enable customized programs to send and receive messages across the PLC network.

Programming the PLC's over the network with the relay ladder diagram chart program in a personal computer enables a programmer to change the set operation of a machine or process as desired. The operation at the personal computer, however, is still one step, the programming step, removed from personally controlling operation of the machine tool or process station. Any manufacturing changes desired by the operator must be implemented through re-programming the individual PLC.

Developing customized programs that directly enable an operator at a personal computer to supervise and actually control the automatic operation of the PLCs quickly becomes expensive due to the large amount of highly skilled labor required. Such custom programs obtain stated information from the PLCs as it occurs and send control information in response to changes in the status information. Such information exchange is described as "real time" to emphasize that there exists little or no delay in receiving the information after the information becomes "real" or available. The cost of updating or varying the customized programs to provide flexibility in the automated operation of an entire plant becomes prohibitive. Further, varying custom programs often reaches

a point where the cost of making the variation within the restraints of the original program exceeds the cost of writing a new custom program. The exchange of "real time" information stands as the key to true operator control of the manufacturing plant.

Programs have become available that strive to provide this real time exchange of information in controlling, monitoring and analyzing processes and instrumentation. Documentation of such an arrangement is found for example in the articles "Lotus Developer Tools" and "Lotus Measure-Data Direct to Lotus Spreadsheets". The user provides a circuit card interface between his process or instruments and the personal computer. The user also writes a device driver program for the personal computer that facilitates the operating system program to communicate with the circuit card interface. The user then installs a commercially available information acquisition program to transfer the desired information between the operating system program and popular and commercially available spreadsheet, data base and analysis programs. The user then exercises control of the manufacturing processes from the spreadsheet, data base or data analysis program through the information acquisition program. This, however, has some undesirable aspects.

From the user's perspective, every transfer of information to and from the spreadsheet, data base or analysis program must occur through menu selected keystroke commands. Sending information to a PLC occurs by transfers to the acquisition program, the operating system program, the device driver program, the interface card, and the network to the addressed PLC registers. Reading from addressed PLC registers requires sending a transmit command to the addressed PLC in the described sending sequence and then sending a read command to the acquisition program to get the transmitted information from reply registers in the interface card through the device driver program, the operating system program and the acquisition program. While the speed of personal domputers reduces all of these transfers to short, almost unnoticeable periods, there are still some things that cannot be done. For example, the calculation of the arguments contained in the cells of a spreadsheet program cannot directly move information between addressable registers of a PLC and the cells of a displayed spreadsheet. Also, writing a device driver for the interface card remains a time consuming and highly skilled task.

Summary of the Invention

The invention as more particurlarly defined in the appended claims furnishes a program or instruction means that operates through a general purpose spreadsheet program to effect information transfers to and from the addressable registers of a PLC and assigned cells of a displayed spreadsheet directly from the spreadsheet program and without transfers through the operating system program or a specially written device driver program. This facilitates a manufacturer configuring a PLC network with a central control of the manufacturing operation from a spreadsheet program operating in a personal computer. Operator control of the PLCs connected to the network occurs through the flexible and easily understood features commercially available in general purpose spreadsheet programs and facilitates true "real time" control and monitoring of the manufacturing plant directly from the displayed spreadsheet of the spreadsheet program.

The invention provides various spreadsheet functions, as would be known to one skilled in the art as "@ functions", including a single @ function WRITE commands in cells in the displayed spreadsheet to transfer information contained in that cell or information contained in relatively addressed cells to be written into desired and addressed PLC registers. A single @ function READ command in cells of the displayed spreadsheet causes the addressed PLC to transmit the information contained in certain addressed registers directly into that cell or relatively addressed cells in the displayed spreadsheet. These @READ and @WRITE functions become executed simply upon recalculating the displayed spreadsheet to effect the information transfers.

The invention also provides menu selected commands for reading or writing blocks of information between addressed PLC registers and ranges of cells in the displayed spreadsheet. These menu selected commands operate through the spreadsheet program free of any acquisition program or device driver program.

The invention also provides for automatic storage of information transmitted to the central controller from the PLCs at regular intervals or as conditions change in the PLCs. These transmissions from the PLCs to the central controller usually are initiated by the PLCs. The invention furnishes an interrupt program that moves the messages received in the interface circuit card to respective memory locations forming a mailbox for each PLC. This prevents sequential messages from one PLC from overwriting the previous message and losing the information content. Thus the sequential messages initiated from one PLC are stored by the present invention in one mailbox area of sequential memory locations. Under an operator selected menu command, the invention moves all of the sequential messages from each PLC into assigned ranges of cells in the displayed spreadsheet.

In particular, the invention involves a system for operating equipment in which a PLC proximate a machine tool or processing station controls the specific operation of that machine tool or processing station. The PLC also is connected to a network of PLCs and is able to receive messages that control operation of the machine tool or processing station and is capable of transmitting messages indicating the condition of the machine tool or processing station. These messages are in the form of electrical signals usually carried on co-axial or twin-axial cable.

A personal computer or processor includes an interface circuit card electrically connected to the data, address and control leads of the personal computer. Usually this interface circuit card rests in an expansion slot of the personal computer. The interface card connects to the network of PLCs and includes addressable registers to which the PLCs can transmit messages. The interface card also includes command and reply registers that can respectively transmit and receive messages to and from the PLCs. The registers of the interface card are accessible by instructions executed in the personal computer.

The personal computer typically includes a visual display, a keyboard, instruction storage memory and a central processor. This central processor executes instructions, in the form of electrical signals, contained in the instruction storage memory in cooperation with characters provided to the displayed and readable by a user and in cooperation with characters received from the user through the keyboard.

This system also includes a general purpose spreadsheet program contained in the instruction storage memory for effecting a general purpose spreadsheet program in the personal computer. This spreadsheet program causes a spreadsheet to be displayed on the visual display. The displayed spreadsheet presents cells into which the user can insert information and @ functions to execute desired arguments. The displayed spreadsheet also provides menu commands selectable by the user to effect desired actions in the displayed spreadsheet, but the spreadsheet program normally is only able to effect movement of information between defined files of data contained in the memory and the cells.

This system further includes add-in instructions for presenting add-in @ functions and add-in menu commands operating through the spreadsheet program and displayed spreadsheet for cell contained @ functions and menu commands to move messages directly between the cells and the interface card with the interface card then moving the messages across the network of PLCs. This obtains a system in which the user can transmit messages controlling the operation of the machine tool or process station directly from the cells of the displayed spreadsheet to the addressed registers of a PLC and can request messages indicating the condition of the machine tool or processing station directly from the cells and receive the status information messages indicating the condition of the equipment directly in the cells.

In another aspect, this system includes add-in instruction means contained in the memory for presenting add-in menu commands and interrupt selected instructions. These add-in menu commands and interrupt selected instructions operate through the spreadsheet program for the interrupt instructions to move sequentially received messages initiated by the PLCs and received in the interface card to respective assigned address locations in memory upon occurrence of an interrupt signal generated by the interface card indicating receipt of such messages. User selection of the add-in menu commands move the message from the locations in memory to respective assigned ranges of cells in the displayed spreadsheet. Messages initiated from the PLC indicating the condition of the machine tools or process stations thus can be saved and moved directly to the cells of the displayed spreadsheet.

The general purpose spreadsheet program in the personal computer becomes initialized for operation with the message transfer from the cells of the displayed spreadsheet through a displayed initialization spreadsheet operating through the spreadsheet program. Add-in instructions contained in the memory provide this initialization spreadsheet and care for the loading of address assignments into certain memory locations accessible by the different parts of the spreadsheet program for effecting and achieving the desired result.

The invention also encompasses a substrate such as a floppy disk ordinarily used in personal computer, carrying the add-in instructions for effecting the transfer of message information to and from the cells of the displayed spreadsheet through the spreadsheet program.

With this invention, an operator can monitor and control the operation of many PLCs on a "real time" basis as conditions change in the manufacturing environment. The spreadsheet program provides many mathematical arguments that can be arranged to perform complex calculations rapidly in determining desired efficiencies and qualities of manufacturing. The ability to obtain "real time" data and give "real time" manufacturing commands in conjunction with these calculations for controlling large manufacturing plants achieves manufacturing control not previously available in an economical and easily understood system.

Brief Description of the Drawings

Figure 1 is a schematic block diagram of a PLC system connected to a personal computer with an add-in program of the invention operating with a general purpose spreadsheet program to furnish real time message transfers between the network and the spreadsheet program;

Figure 2 is a schematic block diagram of the system of the invention in more detail than depicted in Figure 1;

Figure 3 is a schematic block diagram of three aspects of the add-in program of the invention;

Figure 4 is a diagram of the menu commands provided through the add-in program of the invention;

Figure 5 is a schematic block diagram of the interaction between the hardware of a personal computer and the program instructions of a spreadsheet program including the invention to move received messages to assigned memory locations in a mailbox area and to dump messages stored in the mailbox memory locations into assigned ranges of cells in the spreadsheet program;

Figure 6 is a schematic block diagram depicting the interaction between the hardware of a personal computer and a spreadsheet program including the invention executing an @READ function to read information from an addressable register in a PLC to a desired cell in the displayed spreadsheet;

Figure 7 is a schematic block diagram of the interaction between the hardware of a personal computer and a spreadsheet program including the invention executing an @WRITE function to write information from a cell in the displayed spreadsheet to an addressed register in a PLC;

Figure 8 is a schematic block diagram depicting the interaction between the hardware of a programmable computer and the menu commands of a spreadsheet program including the invention to perform a menu selected READ command of reading blocks of information from addressed registers in a PLC to assigned cells in the displayed spreadsheet;

Figure 9 is a schematic block diagram of the interaction between the hardware of a personal computer and a spreadsheet program including the invention to perform a menu selected WRITE command of writing information from assigned cells in a displayed spreadsheet to addressed registers in a PLC;

Figure 10 is a schematic block diagram depicting the interaction between the hardware of a personal computer and a spreadsheet program including the invention to initialize the operation of the spreadsheet program in the personal computer through a displayed spreadsheet;

Figure 11 is a schematic block diagram of the interaction between the hardware of a personal computer and the program instructions of the invention similar to that shown in Figure 5 and further emphasizing the date and time stamping of received messages initiated from the PLCs;

Figure 12 is a schematic block diagram showing the interaction between hardware of a personal computer and program instructions of a spreadsheet program including the invention in initializing pointers to and the contents of memory locations for local variables;

Figure 13 is a schematic block diagram depicting the organization for the add-in instructions of the invention for forming the add-in file SYL.ADN; and

Figure 14 is a schematic block diagram depicting the organization for the add-in instructions of the invention for forming the add-in file SYLAF.ADN.

Description of the Preferred Embodiment

In Figure 1, a plant network 20 of programmable logic controllers (PLCs) to operate equipment stands connected to a personal computer 22 through a circuit card interface 24. The personal computer 22 operates a general purpose spreadsheet program 26 that has no ability to get or give real time messages over the plant network of PLCs. This spreadsheet program can be such as LOTUS 1-2-3 available from Lotus Development Corporation, Cambridge, Massachusetts. The invention furnishes an add-in program 28 that operates through the general purpose spreadsheet program 26 to effect real time message transfers between the plant network of PLCs 20 and cells of the spreadsheet program 26. The transfer of messages through the spreadsheet program to the circuit card interface is depicted by double arrowheaded conductors 30 diagrammatically indicating the message transfer path, and the connection between the plant network 20 and the circuit card 24 is depicted by conductors 32.

In constructing this system 18, the user interconnects all of the PLCs with desired co-axial or twin-axial wires and constructs and arranges all the PLCs to effect positive message transfer between addressed registers of the PLCs. The user then obtains an interface circuit card 24, and installs it in an expansion slot of a personal computer 22. The interface card 24 then becomes connected to the network 20 and is assigned suitable network addresses. The user then loads the general purpose spreadsheet program into the personal computer using disk drives as desired and finally loads the add-in program 28 of the invention into the program files of the spreadsheet program.

The owner of the spreadsheet program known as LOTUS 1-2-3 has provided suitable instructions in the spreadsheet program for accepting the add-in program disclosed herein. The developer of the LOTUS 1-2-3 program also commercially sells program instructions under the trademark LOTUS Developer Tools used by the inventor of the present invention to effect the operation he desired. The LOTUS Developer Tools instruction program enables a programmer to develop instructions that extend or enhance the capability of the spreadsheet program and to create new @ functions to be implemented in a LOTUS 1-2-3 spreadsheet. The program instructions of the LOTUS Developer Tools programs form no part of the invention herein but were used to develop the program instructions used in the invention. After all of this preliminary construction and arrangement of the hardware and program instructions, the user stands ready to program the PLCs and control the operation of the machine tools and process stations that they operate.

In Figure 2, the plant network 20 interface assembly 34 connects to conductors 32 and is assigned address 45. PLC 36 connects to interface assembly 34 to operate machine 38 and is assigned address 045. PLC 40 also connects to interface assembly 34 to operate machine 42 and is assigned address 145. Interface assembly 34 likewise connects to conductors 32 and is assigned address 22. PLC 46 connects to interface assembly 44 to operate machine 48 and is assigned address 022. PLC 50 connects to interface assembly 44 to operate machine 52 and is assigned address 122. Interface assemblies 34 and 44 thus provide hardware connections to

the co-axial or twin-axial conductors 32 and provide the two least significant numbers in the addresses for the PLCs 36, 40, 46 and 50. Conductors 32 connect to interface circuit card 24 at connector 54. In this system, interface card 24 has been assigned address 009 for purposes of explanation. Interface card 24 also can connect to an auxiliary or local PLC 56 through conductors 58 and connector 60. PLC 56 has been assigned 107 and is similar in construction and designed to PLCs 36, 40, 46 and 50. PLC 56 contains sensor and control registers 62 that connect to sensors 64, controls 66 and machine under control 68. The connections between PLCs 36, 40, 46 and 50 to their respective machines parallel the connection between PLC 56 and machine 58. PLC 56, its machine under control 68 and the connection to interface card 24 define what can be called a local system 70.

Interface card 24 includes command, replay and receiving registers 72 and can produce an interrupt signal on leads 74 under conditions to be described.

Personal computer 22 includes RAM memory 76 that contains the spreadsheet program 26, the add-in 28, mailbox memory locations 78 and pointer to local variables memory location 80.

Spreadsheet program 26 presents cells, menu driven commands and @ functions that can be inserted in the cells. Add-in program 28 present three features: menu commands to initialize the features for moving messages between the cells and the PLC registers, block moves of information between the cells and the PLC registers, fixed and floating point information formats and dumping the contents of the mailbox memory locations to assigned cells in the spreadsheet; @ functions to move messages between the spreadsheet cells and the PLC registers, and to perform logical operations; and interrupt driven information transfers from the interface card registers to corresponding memory locations in the mailbox area of the RAM memory 76. The add-in program 28 also includes memory locations 82 for local variables used in effecting the message transfers through the spreadsheet program 26.

Personal computer 22 also includes an operator usable keyboard 84, a visual display 86, a disk drive 88 that can receive a disk 90 carrying the spreadsheet program 26 or the add-in program 28. Disk 90 comprises a substrate capable of carrying program instructions in the form of such as magnetic bits but could be any other kind of substrate desired to carry the program instructions for loading into the memory of the personal computer.

It will be understood that in the arrangement depicted in Figure 2, the add-in program 28 operates through the spreadsheet program 26 to perform the indicated message transfers. After the add-in program becomes installed, the user operates only through spreadsheet @ functions and menu commands to effect the message transfers to the address registers of the PLCs. Alternatively, the plant network 20 of PLCs and the local systems 70 can be arranged as desired to operate desired machine tools and process stations.

Referring to Figure 3, add-in program 28 provides menu commands 92 that the user enters through a keystroke of ALT-F7 on the keyboard of the personal computer. The interrupt instructions 94 perform a mailbox interrupt routine identified in the source code listing herein as MAIL_INT. The interrupt instructions 94 also perform a timer interrupt routine identified in the source code TIMER_INT. The user selectable @ functions insertable in the cells of the displayed spreadsheet include @READ, @WRITE, @BREAD, @BWRITE, @SYLERR, @MAIL, @FPREAD, @FPWRITE, @AND, @OR, @NOT and @TIMEOUT.

These @ functions operate with a particular format of information enclosed in parentheses following the @ function command. The format of information can include an address route specified to be a string of numbers separated by commas identifying first the address of the interface card in the personal computer and secondly the address of the PLC. Following the address can be a register addresses simply stated with a decimal number. Following the register number is a count number stated as a positive decimal number to indicate the number of registers in addition to the identified register from which information is to be read or written. Lastly, a bit number, ranging from 1 to 16 identifies the particular bit in a particular register in a particular PLC to be read to or from the cells.

With this formal of message information, the @READ function requires the address and register information to read one register from the PLC device. The @WRITE function requires the address, register and value to be written to the identified register in the PLC. The values can range from -32768 to +32767. The @BREAD function requires an address, register and bit number format to read the register and return the contents of the identified bit position to the cell. The @BWRITE function requires the address, register, bit number and value to write a particular bit to an addressed register. A value of zero turns the bit off while a value not equal to zero turns the bit on. The @SYLERR function gives the last error message produce by the interface card.

The @MAIL function requires no following message information and obtains the address of the receiving register that received the last network message initiated by a PLC. This will be described presently.

The @FPREAD function requires an address and register identification to read one floating point register from a PLC. The register must be identified with an odd number. This returns a floating point number to a cell. The @FPWRITE function requires an address, register and value to write one floating point number to two registers starting at the identified register value. The register must be odd, and the value must be a floating point number in the range of - 3.4E38 to 3.4E38.

The @TIMEOUT function specifies the maximum time to wait before a response is received from a PLC on the network and requires the specification of time in seconds. The logical functions occur in accordance with

their names.

In Figure 4, the menu 98 appearing upon the user entering the keystrokes ALT-F7 comprise five commands. Initializing the local interface card operates through the INIT routine in the source code. Reading a block of 128 registers from a PLC operate through the routine BLOCK_READ. Writing information to a block of 128 PLC registers operates under the routine BLOCK_WRITE. The floating point command operates through a sub-menu to read or write a block of 32 registers. Reading the block of 32 registers operates through the floating point numbers routine FP_READ. Writing to a block of 32 registers operates under the floating point numbers routine FP_WRITE.

The mailbox command under the ALT-F7 menu operates through two nested sub-menus. The first sub-menu 102 presents three commands. The SET command sets or assigns the mailbox memory locations 78 in the RAM memory 76 that receive information or data from a group of registers in the interface card 24. The SET command sets or assigns the cell locations in the displayed spreadsheet program that are to receive the mailbox data or messages. This command operates through the routine MAIL_SET.

The second command under the first sub-menu 102 is the DUMP command to execute the movement of the contents of the mailbox memory locations from the RAM memory to the assigned cells in the displayed spreadsheet program. The DUMP command operates under the routine MAIL_DUMP.

The third command under the first sub-menu 102 is reset and calls up a sub-sub-menu 104 with two commands in it. The first command of the sub-sub-menu 104 is OUTPUT and changes the displayed spreadsheet cell locations receiving data or information from the DUMP command. The OUTPUT command operates under the routine MBR_OUTPUT. The second command under the sub-sub-menu is a mailbox command. It operates to clear a specified or assigned mailbox from the memory locations in the RAM memory. It operates under the routine of MBR_MB.

The last command under the ALT-F7 menu is to quit the menu.

In Figure 5, network messages addressed to particular receiving registers 106a through 106n in interface circuit card 24 pass over conductors 32. Substantively, the PLCs have been programmed to initiate message transfers to the interface circuit card at address 009 periodically or upon changes in the status of the controlled machine tool or process station. The initiation of sending the messages becomes programmed into the PLCs under operator control. Each PLC addresses a particular one of the receiving registers 106a through 106n for receiving the sent message. Alternatively, the interface circuit card 24 can have one receiving register to receive all messages initiated by the PLCs upon suitable arrangement of the system to insure positive receipt and protection of the messages. In this system, interrupt-call program 94 moves the messages from the receiving registers to the respective assigned mailbox memory locations 78a through 78n while adding a time/date identifying information with each message. This later identifies to the spreadsheet program and the user the relative time of the status message from the PLCs. In moving the messages from the registers to the mailbox memory location, the interrupt-call program 94 will move the sequential messages received in receiving register 106a into sequential memory locations following memory locations 78a. The user assigned the mailbox memory locations 78a under control of the set mailbox location command in sub-menu 102 depicted in Figure 4. Each mailbox can be as large as desired by the user to retain the sequential messages and time/date stamp of the messages until they are moved into the displayed spreadsheet.

Menu selected DUMP command 108 can be executed when desired by the user to move the contents of each mailbox in memory locations 78a through 78n to respective and assigned ranges of cells 110a through 110n in displayed spreadsheet 112. Thus the sequential messages and time/date stamps in memory locations 78a comprising one mailbox are moved by the DUMP command routine 108 moving them to sequential cells 110a in the displayed spreadsheet 112. In this manner, sequential message from the PLCs received in receiving register 106a are transferred to the mailbox memory locations 78a and then are dumped into the range of cells 110a. Likewise, PLC initiated messages to receiving register 106n are moved under interrupt program control to the mailbox at memory location 78n. The DUMP command routine moves the messages in mailbox memory location 78n into the range of cells 110n. This neatly provides organized and assigned transfer of PLC initiated messages to the displayed spreadsheet to insure positive message transfer to the cells and to insure against loss of PLC initiated messages.

In Figure 6, spreadsheet program with add-in features 26, 28 identified in one block because they operate together as one unit and shall be identified with the double reference characters 26, 28. Spreadsheet program 26, 28 presents another spreadsheet screen 14 including an @READ function in cell C4. In cell C4, the @ function @READ ("route 009, 145", 28) indicates to the command registers 116 that a message is to be sent across the network conductors 32. The message commands PLC address 145 to send the contents of its register 28 back to the interface circuit card address 009, and thereafter the message received from the PLC address 145, register 28 is to be placed in cell C4 of the spreadsheet screen 114. Upon calculation of the spreadsheet screen 114, or the portion of the spreadsheet screen including cell C4, this @READ command becomes executed. The spreadsheet program 26, 28 operates to tell the interface circuit card 24 to send and receive the appropriate messages through respective command registers 116 and reply registers 118. Upon receipt of the message con-

taining the contents of PLC address 145, register 28, the spreadsheet program 26, 28 obtains the contents of the reply register 118 and moves it into the cell C4 from which the user can observe the number 100.

In Figure 7, spreadsheet program 26, 28 presents another spreadsheet screen 120 having a cell E4 containing an @WRITE function. That function is @WRITE ("route 009, 022", 56, 300). It indicates that interface circuit card 24, address 009 is to write a message to PLC address 022, register 56 with the message containing the contents of cell E4 which is 300. Upon executing the spreadsheet screen 120 or that portion containing cell E4, spreadsheet program 26, 28 conveys the necessary information to command register 116 and interface circuit card 24 to effect the desired message. That message then travels across the conductors 32 to PLC 46 having address 022 and register 56 to insert the number 300 therein.

The @ functions depicted in Figures 6 and 7 thus can be used upon normal recalculation of the spreadsheet screens 114 and 120 to transfer information between the cells of the spreadsheet screen and registers of desired PLCs. The user does not have to prompt individual message transfers and the user can write the mathematical and logical @ functions available in the general purpose spreadsheet program 26 to establish procedures for automatically controlling operations of such as machines 42 and 48. Specifically, the @READ function depicted in Figure 6 can indicate the number of screws manufactured on an automatic screw machine 42. When that number equals 100, the mathematical and logical @ functions inserted into such as spreadsheet screen 120 by the user can write the value 300 into PLC address 022, register 56 to advance a conveyor machine 48 to the next work station.

In Figure 8, spreadsheet program 26, 28 presents another spreadsheet screen 122 used in conjunction with the menu selected READ command 124 to read a block of PLC registers into a range of cells in spreadsheet 122. The READ command 124 first prompts the user to enter the route, register and count range of PLC registers to be read, respectively in cells A2, B2 and C2. Thus the message route is 009, 145. The register to read starts at register 28, and the range of PLC registers thereafter is three. The READ command 124 also prompts the user to enter the output range of cells to receive the contents of the PLC registers. In spreadsheet screen 122, the user has entered the cells to be A5, B5 and C5. Upon executing the menu READ command 124, spreadsheet program 26, 28 moves appropriate message information to command registers 116 which sends appropriate messages over conductors 32. PLC 40 having address 145 receives the READ command and transmits messages containing the contents of registers 28, 29 and 30 back to the interface circuit card 24 having address 009. Substantively, registers 28, 29 and 30 in PLC 40 can represent the status of screw machine 42. When reply registers 118 receive the messages containing the contents of registers 28, 29 and 30, spreadsheet program 26, 28 moves the information content thereof respectively into cells A5, B5 and C5.

In Figure 9, spreadsheet program 26, 28 presents a new spreadsheet screen 126 used to write a block of data to the registers of a PLC. The menu selected write command 128 first prompts the user to enter the route, register and count range of the PLC registers to be written to. This information contained in spreadsheet screen 126, cells A2, B2 and C2 respectively. The WRITE command 128 also prompts the user to enter the range of cells, the contents of which are to be written to the PLC registers. This information is contained respectively in cells A5, B5 and C5. Upon executing the WRITE command 128, spreadsheet program 26, 28 moves the appropriate information to command registers 116 of interface circuit card 24. Interface card 24 then sends the appropriate messages across the conductors 32 to PLC 40, having address 145, and particularly to registers 5, 6 and 7. This information can be such as control information to control the operation of such as automatic screw machine 42.

In this manner, the menu selected READ and WRITE commands can monitor the status of a machine and control its operation through menu selected commands and spreadsheet screen contained messages address and content information.

In Figure 10, spreadsheet program 26, 28 presents an initialization spreadsheet screen 130 used to initialize local variables and information in the interface circuit card 24 before messages transfers can occur. Interface circuit card 24 contains dip switches that are user setable to assign an address of the personal computer to the registers of the interface circuit card 24. In the example depicted in Figure 10, this address is selected to be CA000 expressed in hexadecimal. Circuit card 24 also contains control information registers for receiving and containing a network address 132, the number of PLCs on the network 134, the speed at which the network operates 136 and the interrupt signal 138 produced by the circuit card 24 on lead 74, previously explained. Spreadsheet screen 130 contains respectively in cells C2 through C6 the information inserted by the user to initialize the spreadsheet program 26, 28 and the circuit card 24. Upon execution of a menu command, spreadsheet program 26, 28 moves this information of spreadsheet screen 130 to the circuit 24 and to two local variables memory locations 140 and 142. As depicted in Figure 10, the memory locations for the two indicated local variable occur within the spreadsheet program 26, 28 for easy access thereto. In particular, cell C9 indicates that the network address for card 24 shall be 009. Cell C3 indicates that the number of PLCs in the network is 100. Cell C4 indicates that the speed of the messages on the network is indicated by the code 62. Cell C6 indicates that the mailbox interrupt signal shall be number 2. These four pieces of information respectively are moved to registers 132 through 138 and circuit card 24 upon execution of a

menu selected command, such as INITIALIZE invoked by ALT-F7. Additionally, cell C5 indicates that the address of the circuit card 24 in the expansion slot of the personal computer has been set to be CA000 and hexadecimal, and this information is moved by the spreadsheet program 26, 28 to the NIM_SEG local variable memory location 140. The mailbox interrupt number 2 also is moved by the spreadsheet program 26, 28 to the interrupt local variable memory location 142. The contents of cells D8 and D9 indicate to the spreadsheet program 26, 28 that subsequent selections of the spreadsheet program in the personal computer automatically can initialize the spreadsheet program 26, 28 to the values indicated in cells C2 through C6 and that it can automatically attach the @ functions into the spreadsheet program 26, 28.

In Figure 11, the interrupt call program 94 operates to move messages from the interface circuit card 24 into the assigned mailbox memory locations in mailbox memory locations 78 together with the time and date information. Interrupt call 94 directly operates to move the operating system time 144 into memory location A, expressed in hexadecimal. A periodic hardware interrupt 146 occurs in the personal computer at regular intervals. The timer interrupt routine, TIMER_INT 148 increments a date variable and places same in a memory location 150 of a "date" variable among the local variables of the spreadsheet program 26, 28.

Previously, a menu selected mailbox SET command 152 is used to effect a routine 154 that assigns memory locations for the mailbox and that effects a routine 156 to initialize a start-date variable from the operating system date. Routine 156 places the "start-date" variable in memory location 158 of the local variables.

The operating system displays the date by using an offset from a fixed date such as January 1, 1900. During the initialization of the spreadsheet program 26, 28, through the menu selected mailbox command 152, the add-in commands 28 ascertain the offset from that initial date to the present date and insert and indication of same in the "start-date" variable memory location 158. The operating system normally does not provide the date or date offset in a manner accessible from the spreadsheet program with add-in features 26, 28 so instead, the invention provides the timer interrupt routine 148. This counts the periodic hardware interrupts occurring in the personal computer at 18 times a second to increment a "date" variable and place it in the memory location 150. Thus the date column of the memory location of the mailbox carries an offset from the initial date of the operating system to the present date. Later, the menu selected dump command 108 can use the offset with the start-date variable properly to format the date in such as cell A3 of the displayed spreadsheet 112. The operating system time, which is available directly from the operating system time to the spreadsheet program with add-in features 26, 28 rests in the time column of the memory location of the mailbox and can be transformed upon the effecting the menu selected DUMP command 108 to present the proper time at such as cell B3 in the displayed spreadsheet 112. The date format and time format functions become effected through blocks 160 and 162 respectively.

As previously explained, the menu selected DUMP command 108 moves the messages from the respective mailboxes in the memory location 78 into the assigned output cells of the displayed spreadsheet 112. In the depicted example of Figure 11, the output ranges of cells are A3 through E3.

In Figure 12, the add-in program 12 has been assigned to memory locations 164 and 168. These can be sequential or non-sequential as desired by the add-in features of the spreadsheet program 26. In any event, memory locations 164 contain the program named SYLAF.ADN for the @ functions while memory locations 168 contain the program called SYL.ADN for the menu and interrupt commands. Memory locations 168 also include the memory locations 82 of the local variables such as IPC.NIM_SEG, IPC.ACTIVE and IPC.TIMEOUT_SECS.

When the spreadsheet program 26 loads the add-in files into the memory locations, it first loads the SYL.ADN file into memory location 168. After loading, the spreadsheet program 26 executes the start-up init routine 168. That routine 168 executes block 170 to clear the pointers to the locations of the local variables contained in memory locations 80. Memory locations 80 comprises an inter-program communications areas containing pointers to the memory locations of local variables. The memory location of the pointer is defined to be addresses 04F8 hexadecimal and 04FA hexadecimal. The first address carries the offset of the pointer while the second address carries segment of the pointer.

When the spreadsheet program 26 next loads the memory locations 164 with the program SYLAF.ADN, the start-up routine 172 contained therein sets an error if the pointers are not cleared. That occurs at block 174.

The next thing that occurs is for menu selected initialized routine 176 under memory locations 166 to set the pointers to the locations of the local variables at block 178. The menu selected initialized routine 176 also sets the contents of the local variables in block 180 as has been described.

Thereafter, menu selected function routines 182 get and use the contents of local variables through block 184 with the variables being contained within the file or program of SYL.ADN which also contains the menu selected function routine. Thus the menu selected function routines do not need the offset pointer to the local variables but can access them directly.

The @ function routines, however, can be located other than sequentially with the local variables and get the pointer to the local variables through block 186. Once the @ function routines 188 have the pointers to the local variables, they can obtain the information contained therein to execute the desired message transfers previously described.

In Figure 13, master file 190 for the menu functions SYL.ASM includes files SYLI.ASM, SYLHELP.ASM, SYLINT.ASM, SYLMB.ASM, SYLMDATE.ASM, SYLERR.ASM, SYLADDR.ASM, and SYLFP.ASM. These include files perform functions identified in block 190. These menu function files of master file 190 become assembled in block 192 to obtain the add-in file SYL.ADN in block 194.

In Figure 14, master file 196 for the @ functions includes the SYLAF.ASM file and include files SYLFPAT.ASM and SYLNORM.ASM. The include files in master file 196 perform the identified functions. These files are assembled in block 198 to obtain the add-in file SYLAF.ADN in block 200.

The source code listing for the add-in program 28 of the invention comprises program pages 1-118 and is set forth herein. The invention thus achieves a user friendly procedure for moving information messages to and from cells in a spreadsheet program and addressed registers in PLCs. This provides a user with easy access to the real-time information needed to perform real-time control of machinery and processing stations.

In the preferred embodiment, the PLC network 20 is that made and sold by the Square D Company of Palatine, Illinois, under the SY/NET trademark of programmable logic controllers, network interface modules and SY/LINK interface cards for personal computers. The PLCs in this network are identified with the mark SY/MAX. Network specifications for all of this hardware and relay ladder programming instructions are commercially available.

Modifications and variations of the present inventions are possible in light of the above teachings. It will be understood that different commercial versions of spreadsheet programs may use variations of the names used herein for functions used or inserted in the cells. For example, the @ functions can also be called functions without the @ prefix.

Claims

1. A system for providing real time control of operating equipment (42) with a programmable logic controller (70) said programmable logic controller (70) being capable of receiving control messages to control operation of said equipment (42) and being capable of transmitting status messages indicating a condition of said equipment (42), said system comprising, a computer (22) having an operably interconnected visual display (86), a keyboard (84), a memory (76) and a central processor, said computer (22) for sequentially executing instructions contained in said memory (76) in cooperation with characters received from said keyboard (84), an interface (24) electrically connected between said computer (22) and said programmable logic controller (70) for transmitting messages between said programmable logic controller (70) and said computer (22), and spreadsheet instruction means (26) contained in said memory (76) for effecting a general purpose spreadsheet program in said computer (22), said spreadsheet instruction means (26) providing cells displayable by said visual display (86), into which a user can insert information, spreadsheet functions that can be inserted into said cells via said keyboard (84) and selectable menu commands to effect actions in said spreadsheet instruction means (26), said spreadsheet instruction means (26) normally only being able to effect movement of information between files of data contained in said memory (76) and said cells, characterized by add-in instruction means (28) contained in said memory (76) for providing add-in sDreadsheet functions (96) and add-in menu commands (92) operating through said spreadsheet instruction means (26), said add-in instruction means (28) for moving messages directly between said cells and said programmable logic controller (70) via said interface (24), said messages moving from said cells to said programmable logic controller (70) comprising said control messages for controlling the operation of said equipment (42) and said messages moving from said programmable logic controller (70) to said cells comprising said status messages indicating the condition of said equipment (42).

2. The system of claim 1, characterized by multiple programmable logic controllers (70), said interface (24) electrically connected between said computer (22) and said multiple programmable logic controllers (70) for receiving messages from said multiple programmable logic controllers (70), said interface (24) including at least one message register for each programmable controller (70), the messages from each of said programmable logic controller (70) being overwritten on a preceding message in said registers, said interface (24) including interrupt means for interrupting operation of said computer (22) in response to receiving a message in any one of said message registers, and said add-in instruction means (28) for presenting interrupt selected instructions operating through said spreadsheet instruction means (26) for said interrupt selected instructions to move sequentially received messages from said message registers to respective assigned address locations in said memory (76) upon occurrence of each interrupt signal, and for said menu commands to move said messages from said assigned address locations in said memory (76) to respective assigned cells in said spreadsheet instruction means (26) so that messages from said programmable logic controllers (70) indicating the condition of said equipment (42) can be saved and moved directly to said cells.

3. The system of claim 1 or 2, characterized in that said programmable logic controller (70) including addressable registers, said addressable registers being capable of receiving said control messages, and being capable of transmitting said status messages, said message registers capable of transmitting addressed messages to said addressable registers in said programmable logic controller (70) and being accessible by said computer (22), said interface (24) for transmitting addressed messages to said addressable registers in said programmable logic controller (70) in response to message information received from said computer (22), and for presenting to said computer (22) the information contained in receiving addressed messages from said addressable registers in said programmable logic controller (70).

4. The system of claim 1 characterized in that said add-in instruction means (28) includes add-in read means for causing said programmable logic controller (70) to transmit a status message to said spreadsheet instruction means (26).

5. The system of claim 4 characterized in that said add-in read means includes characters indicating an spreadsheet read function and route addresses of said interface (24) and said programmable logic controller (70).

6. The system of claim 5 characterized in that said add-in read means includes characters indicating a certain register in said programmable logic controller (70), the contents of which are to be sent in said status message.

7. The system of claim 4 characterized in that said cells of said spreadsheet instruction means (26) are addressable by characters from said keyboard (84) and said add-in read means are addressed into a certain cell by characters from said keyboard (84).

8. The system of claim 1 characterized in that said add-in instruction means (28) includes add-in write means for transmitting a control message to said programmable logic controller (70).

9. The system of claim 8 characterized in that said add-in write means includes characters indicating a spreadsheet write function and route addresses of said interface (24) and said programmable logic controller (70).

10. The system of claim 9 characterized in that said add-in write means includes characters indicating a certain register in said programmable logic controller (70) that is to receive said control message.

11. The system of claim 1 characterized in that said add-in instruction means (28) includes menu initialize means for moving initial system information to said interface (24).

12. The system of claim 1 characterized in that said add-in instruction means (28) includes first file means for containing local variables used to move messages directly between said cells and said interface (24) and for containing said menu commands, said add-in instruction means (28) also including second file means for containing said add-in spreadsheet functions (96), and said spreadsheet instruction means (26) including assignment means for assigning said first and second file means to specific memory locations.

13. A process for providing real time control of operating equipment (42) with a programmable logic controller (70), said programmable logic controller (70) being capable of receiving messages to control operation of said equipment (42) and being capable of transmitting messages indicating a condition of said equipment (42), said process comprising: providing a computer (22) having an operably interconnected visual display (86), a keyboard (84), a memory (76) and a central processor, said providing including causing said computer (22) sequentially to execute instructions, contained in said memory (76) in cooperation with characters received from said keyboar (84); furnishing an interface (24) electrically connected between said computer (22) and said programmable logic controller (70) capable of transmitting messages between said programmable logic controller (70) and said computers (22); effecting a general purpose spreadsheet program in said computer (22), said spreadsheet program providing cells displayable by said visual display (86) into which a user can insert information, spreadsheet functions that can be inserted into said cells via said keyboard (84) and selectable menu commands to effect actions in said spreadsheet program, said spreadsheet program normally only being able to effect movement of information between files of data contained in said memory (76) and said cells, characterized by effecting add-in spreadsheet functions (96) and add-in menu commands (92) in said computer operating through said spreadsheet program, to move messages directly between said cells and said programmable logic controller (70) via said interface (24) said messages moving from said cells to said programmable logic controller comprising said control messages for controlling the operation of said equipment (42) and said messages moving from said programmable logic controller (70) to said cells comprising said status messages indicating the condition of said equipment (42) and can request messages indicating the condition of said equipment

(42) directly from said cells and receive said messages indicating the condition of said equipment (42) directly in said cells.

Patentansprüche

1. System zur Bereitstellung einer Echtzeitsteuerung einer Betriebsanlage (42) mit einem programmierbaren Logiksteuergerät (70), wobei das programmierbare Logiksteuergerät (70) Steuernachrichten zur Steuerung der Betriebsweise der Anlage (42) empfangen und Zustandsnachrichten, die einen Zustand der Anlage (42) anzeigen, übertragen kann, wobei das System folgendes umfaßt: einen Computer (22) mit einer funktionsfähig damit verbundenen Sichtanzeige (86), einer Tastatur (84), einem Speicher (76) und einem Zentralprozessor, wobei der Computer (22) sequentiell in dem Speicher (76) vorhandene Befehle zusammenwirkend mit von der Tastatur (84) empfangenen Zeichen ausführt; eine Schnittstelle (24), die elektrisch zwischen den genannten Computer (22) und das programmierte Logiksteuergerät (70) geschaltet ist, um Nachrichten zwischen dem programmierbaren Logiksteuergerät (70) und dem Computer (22) zu übertragen, und wobei eine Kalkulationstabellen-Befehlseinrichtung (26) in dem Speicher (76) vorhanden ist, um ein universales Kalkulationsprogramm in dem Computer (22) vorzusehen, wobei die genannte Kalkulationstabellen-Befehlseinrichtung (26) durch die Sichtanzeige (86) anzeigbare Zellen vorsieht, in die ein Anwender Informationen einfügen kann, und wobei die Kalkulationsfunktionen über die Tastatur (84) und auswählbare Menüfunktionen in die Zellen eingefügt werden können, um Vorgänge in der Kalkulationstabellen-Befehlseinrichtung (26) zu bewirken, wobei die Kalkulationstabellen-Befehlseinrichtung (26) normalerweise nur die Übertragung von Informationen zwischen in dem Speicher (76) und in den Zellen enthaltenen Datendateien ausführen kann, gekennzeichnet durch eine Add-In-Befehlseinrichtung (28), die in dem Speicher (76) vorhanden ist, um Add-In-Kalkulationsfunktionen (96) und Add-In-Menübefehle (92) vorzusehen, die durch die Kalkulationstabellen-Befehlseinrichtung (26) ausgeführt werden, wobei die Add-In-Befehlseinrichtung (28) dazu dient, Nachrichten direkt über die genannte Schnittstelle (24) zwischen den genannten Zellen und dem genannten programmierbaren Logiksteuergerät (70) zu übertragen, wobei die Nachrichten von den Zellen zu dem programmierbaren Logiksteuergerät (70) übertragen werden, das die Steuernachrichten zur Steuerung der Funktionsweise der Anlage (42) umfaßt, und wobei die Nachrichten von dem programmierbaren Logiksteuergerät (70) zu den Zellen übertragen werden, die die Zustandsnachrichten aufweisen, die den Zustand der Anlage (42) anzeigen.

2. System nach Anspruch 1, gekennzeichnet durch eine Mehrzahl von programmierbaren Logiksteuergeräten (70), wobei die genannte Schnittstelle (24) elektrisch zwischen den Computer (22) und die Mehrzahl von programmierbaren Logiksteuergeräten (70) geschaltet ist, um Nachrichten von der Mehrzahl von Logiksteuergeräten (70) zu empfangen, wobei die Schnittstelle (24) mindestens ein Nachrichtenregister für jedes programmierbare Logiksteuergerät (70) aufweist, wobei die Nachrichten von jedem programmierbaren Logiksteuergerät (70) über eine vorherige Nachricht in den Registern geschrieben werden, wobei die Schnittstelle (24) eine Unterbrechungseinrichtung umfaßt, um den Betrieb des Computers (22) als Reaktion auf den Empfang einer Nachricht in einem der Nachrichtenregister zu unterbrechen, und wobei die Add-In-Befehlseinrichtung (28) durch die Unterbrechung ausgewählte Befehle vorsieht, die durch die Kalkulationstabellen-Befehlseinrichtung (26) für die durch die Unterbrechung ausgewählten Befehle so ausgeführt werden, daß sequentiell empfangene Nachrichten aus den Nachrichtenregistern bei Eintreten jedes Unterbrechungssignals an entsprechend zugewiesene Adreßplätze in dem Speicher (76) übertragen werden, und wobei die Menübefehle die Nachrichten von den genannten zugewiesenen Adreßplätzen in dem Speicher (76) in entsprechende zugewiesene Zellen in der Kalkulationstabellen-Befehlseinrichtung (26) übertragen werden, so daß Nachrichten aus den programmierbaren Logiksteuergeräten (70), die den Zustand der Anlage (42) anzeigen, gespeichert und direkt in die Zellen übertragen werden können.

3. System nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß das genannte Logiksteuergerät (70) adressierbare Register aufweist, wobei die adressierbaren Register die genannten Steuernachrichten empfangen und die genannten Zustandsnachrichten übertragen können, wobei die genannten Nachrichtenregister adressierbare Nachrichten in die adressierbaren Register in dem programmierbaren Logiksteuergerät (70) übertragen können, und wobei auf sie durch den Computer (22) zugegriffen werden kann, wobei die genannte Schnittstelle (24) als Reaktion auf von dem Computer (22) empfangene Nachrichteninformationen adressierte Nachrichten in die adressierbaren Register in dem programmierbaren Logiksteuergerät (70) übertragen können, und wobei die Schnittstelle dem Computer die Informationen vorsieht, die in den empfangenen adressierbaren Nachrichten aus den adressierbaren Registern in dem programmierbaren Logiksteuergerät (70) enthalten sind.

4. System nach Anspruch 1, dadurch gekennzeichnet, daß die genannte Add-In-Befehlseinrichtung (28) eine Add-In-Leseeinrichtung aufweist, die dafür sorgt, daß das programmierbare Logiksteuergerät (70) eine Zustandsnachricht an die Kalkulationstabellen-Befehlseinrichtung (26) überträgt.

5. System nach Anspruch 4, dadurch gekennzeichnet, daß die genannte Add-In-Leseeinrichtung Zeichen aufweist, die eine Kalkulations-Lesefunktion und Leitadressen der Schnittstelle (2) und des programmierbaren Logiksteuergerätes (70) anzeigen.

6. System nach Anspruch 5, dadurch gekennzeichnet, daß die genannte Add-In-Leseeinrichtung Zeichen aufweist, die ein bestimmtes Register in dem programmierbaren Logiksteuergerät (70) anzeigen, dessen Inhalt in einer Zustandsnachricht übertragen wird.

7. System nach Anspruch 4, dadurch gekennzeichnet, daß die Zellen der Kalkulationstabellen-Befehlseinrichtung (26) durch die Zeichen der Tastatur (84) adressiert werden können, und wobei die genannte Add-In-Leseeinrichtung durch Zeichen von der Tastatur (84) in eine bestimmte Zelle adressiert wird.

8. System nach Anspruch 1, dadurch gekennzeichnet, daß die genannte Add-In-Befehlseinrichtung (28) eine Add-In-Schreibeinrichtung zur Übertragung einer Steuernachricht zu dem programmierbaren Logiksteuergerät (70) umfaßt.

9. System nach Anspruch 8, dadurch gekennzeichnet, daß die genannte Add-In-Schreibeinrichtung Zeichen umfaßt, die eine Kalkulations-Schreibfunktion und Leitadressen der Schnittstelle (24) und des programmierbaren Logiksteuergeräts (70) anzeigen.

10. System nach Anspruch 9, dadurch gekennzeichnet, daß die genannte Add-In-Schreibeinrichtung Zeichen umfaßt, die ein bestimmtes Register in dem programmierbaren Logiksteuergerät (70) anzeigen, das die genannte Steuernachricht empfangen soll.

11. System nach Anspruch 1, dadurch gekennzeichnet, daß die genannte Add-In-Befehlseinrichtung (28) eine Menü-Initialisierungseinrichtung zur Übertragung der anfänglichen Systeminformationen an die Schnittstelle (24) umfaßt.

12. System nach Anspruch 1, dadurch gekennzeichnet, daß die genannte Add-In-Befehlseinrichtung (28) eine erste Dateieinrichtung umfaßt, die lokale Variablen aufweist, die dazu verwendet werden, Nachrichten direkt zwischen den Zellen und der Schnittstelle (24) zu übertragen, und um die genannten Menübefehle aufzuweisen, wobei die genannte Add-In-Befehlseinrichtung (28) ferner eine zweite Dateieinrichtung umfaßt, die die genannten Add-In-Kalkulationsfunktionen (96) aufweist, und wobei die genannte Kalkulationstabellen-Befehlseinrichtung (26) eine Zuweiseungseinrichtung umfaßt, um die genannte erste und zweite Dateieinrichtung bestimmten Speicherplätzen zuzuweisen.

13. Verfahren zur Bereitstellung einer Echtzeitsteuerung einer Betriebsanlage (42) mit einem programmierbaren Logiksteuergerät (70), wobei das programmierbare Logiksteuergerät (70) Nachrichten zur Steuerung der Betriebsweise der Anlage (42) empfangen und Nachrichten, die einen Zustand der Anlage (42) anzeigen, übertragen kann, wobei das Verfahren folgendes umfaßt: Bereitstellung eines Computers (22), der eine funktionsfähig damit verbundene Sichtanzeige (86), eine Tastatur (84), einen Speicher (76) und einen Zentralprozessor aufweist, wobei die Bereitstellung ferner dafür sorgt, daß der Computer (22) sequentiell in dem Speicher (76) vorhandene Befehle ausführt, und zwar zusammenwirkend mit von der Tastatur (84) empfangenen Zeichen; Bereitstellung einer Schnittstelle (24), die elektrisch zwischen den genannten Computer (22) und das programmierte Logiksteuergerät (70) geschaltet ist, um Nachrichten zwischen dem programmierbaren Logiksteuergerät (70) und dem Computer (22) zu übertragen; Bewirken eines universalen Kalkulationsprogramms in dem genannten Computer (22), wobei das Kalkulationsprogramm durch die Sichtanzeige (86) anzeigbare Zellen vorsieht, in die ein Anwender Informationen einfügen kann, und mit Kalkulationsfunktionen, die über die Tastatur (84) in die Zellen eingefügt werden können, und mit auswählbaren Menüfunktionen, die Vorgänge in dem Kalkulationsprogramm bewirken, wobei das Kalkulationsprogramm normalerweise nur die Übertragung von Informationen zwischen in dem Speicher (76) und in den Zellen enthaltenen Datendateien ausführen kann, gekennzeichnet durch die Ausführung von Add-In-Kalkulationsfunktionen (96) und Add-In-Menübefehlen (92) in dem Computer, die durch das Kalkulationsprogramm ausgeführt werden, um Nachrichten direkt über die genannte Schnittstelle (24) zwischen den genannten Zellen und dem genannten programmierbaren Logiksteuergerät (70) zu übertragen, wobei die Nachrichten von den Zellen zu dem programmierbaren Logiksteuergerät übertragen werden, das die Steuernachrichten zur Steuerung der Funktionsweise der Anlage (42) umfaßt, und wobei die Nachrichten, die von dem programmierbaren Logiksteuergerät (70) zu den Zellen übertragen werden, die Zustandsnachrichten umfassen, die den Zustand der Anlage (42) anzeigen, und wobei Nachrichten, die den Zustand der Anlage (42) anzeigen, direkt von den Zellen angefordert werden können, und

wobei Nachrichten, die den Zustand der Anlage (42) anzeigen, direkt in den Zellen empfangen werden können.

Revendications

1. Système destiné à réaliser la commande en temps réel d'un équipement (42) avec un régisseur logique programmable (70) ledit régisseur logique programmable (70) étant capable de recevoir des messages de commande pour commander le fonctionnement dudit équipement (42) et étant capable de transmettre des messages d'état indiquant un état dudit équipement (42), ledit système comprenant un ordinateur (22) pourvu d'un dispositif d'affichage visuel (86), d'un clavier (84), d'une mémoire (76) et d'une unité de traitement centrale interconnectés à l'état opérationnel, ledit ordinateur (22) étant destiné à exécuter de manière séquentielle des instructions contenues dans ladite mémoire (76) en coopération avec des chaînes de caractères reçues dudit clavier (84), d'une interface (24) reliée électriquement entre ledit ordinateur (22) et ledit régisseur logique programmable (70) pour la transmission de messages entre ledit régisseur logique programmable (70) et ledit ordinateur (22), et un moyen de traitement de feuille de calcul (26) contenu dans ladite mémoire (76) destiné à exécuter un programme de gestion de feuilles de calcul d'usage général dans ledit ordinateur (22), ledit moyen de traitement de feuille de calcul (26) disposant de cellules pouvant être affichées par ledit dispositif d'affichage (86), dans lequel un utilisateur peut insérer une information, des fonctions de feuille de calcul qui peuvent être insérées dans lesdites cellules par l'intermédiaire dudit clavier (84) et des instructions de menu pouvant être sélectionnées afin de réaliser des actions dans ledit moyen de traitement de feuille de calcul (26), ledit moyen de traitement de feuille de calcul (26) étant normalement seulement capable de réaliser des déplacements d'informations entre les fichiers de données contenues dans ladite mémoire (76) et lesdites cellules, caractérisé par le fait qu'un moyen de traitement supplémentaire (28) est contenu dans ladite mémoire (76) pour produire des fonctions de feuille de calcul supplémentaires (96) et des instructions de menu supplémentaires (92) fonctionnant par l'intermédiaire dudit moyen de traitement de feuille de calcul (26), ledit moyen de traitement supplémentaire (28) destiné au déplacement des messages directement entre lesdites cellules et ledit régisseur logique programmable (70) par l'intermédiaire de ladite interface (24), lesdits messages se déplaçant depuis lesdites cellules vers ledit régisseur logique programmable (70) comprenant lesdits messages de commande destinés à commander le fonctionnement dudit équipement (42) et lesdits messages se déplaçant entre ledit régisseur logique programmable (70) vers lesdites cellules comprenant lesdits messages d'état indiquant l'état dudit équipement (42).

2. Système selon la revendication 1, caractérisé par des régisseurs logiques programmables multiples (70), ladite interface (24) étant reliée électriquement entre ledit ordinateur (22) et lesdits régisseurs logiques programmables multiples (70) pour recevoir des messages issus desdits régisseurs logiques programmables multiples (70), ladite interface (24) incluant au moins un registre de messages pour chaque régisseur programmable (70), les messages issus de chacun desdits régisseurs logiques programmables (70) recouvrant un ancien message dans lesdits registres, ladite interface (24) incluant un moyen d'interruption pour interrompre le fonctionnement dudit ordinateur (22) en réponse à la réception d'un message dans l'un quelconque desdits registres de messages, et ledit moyen de traitement supplémentaire (28) permettant de disposer d'instructions sélectionnées par interruption fonctionnant par l'intermédiaire dudit moyen de traitement de feuille de calcul (26) pour lesdites instructions sélectionnées par interruption afin de déplacer de manière séquentielle les messages reçus depuis lesdits registres de messages vers des emplacements d'adresses respectifs affectés dans ladite mémoire (76) lors de l'apparition de chaque signal d'interruption, et pour lesdites instructions de menu afin de déplacer lesdits messages entre lesdits emplacements d'adresses affectés dans ladite mémoire (76) vers les cellules respectives affectées dans ledit moyen de traitement de feuille de calcul (26) de façon que des messages issus desdits régisseurs logiques programmables (70) indiquant l'état dudit équipement (42) puissent être sauvegardés et déplacés directement dans lesdites cellules.

3. Système selon la revendication 1 ou 2, caractérisé en ce que ledit régisseur logique programmable (70) incluant des registres adressables, lesdits registres adressables étant capables de recevoir lesdits messages de commande, et étant capables de transmettre lesdits messages d'état lesdits registres de messages pouvant transmettre des messages adressés vers lesdits registres adressables dans ledit régisseur logique programmable (70) et dont l'accès peut être obtenu par l'ordinateur (22), ladite interface (24) destinée à transmettre des messages adressés vers lesdits registres adressables dans ledit régisseur logique programmable (70) en réponse au message l'information reçu dudit ordinateur (22), et pour présenter audit ordinateur (22) l'information contenue en recevant des messages adressés issus desdits registres adressables dans ledit régisseur logique programmable (70).

4. Système selon la revendication 1, caractérisé en ce que ledit moyen de traitement supplémentaire (28) inclut un moyen de lecture supplémentaire pour faire en sorte que ledit régisseur logique programmable (70) émette un message d'état vers ledit moyen de traitement de feuille de calcul (26).

5. Système selon la revendication 4, caractérisé en ce que ledit moyen de lecture supplémentaire inclut une chaîne de caractères désignant une fonction de lecture de feuille de calcul et des adresses de chemin de ladite interface (24) et dudit régisseur logique programmable (70).

6. Système selon la revendication 5, caractérisé en ce que ledit moyen de lecture supplémentaire inclut une chaîne de caractères désignant un certain registre dans ledit régisseur logique programmable (70), dont le contenu doit être envoyé dans ledit message d'état.

7. Système selon la revendication 4, caractérisé en ce que lesdites cellules dudit moyen de traitement de feuille de calcul (26) peuvent être adressées par des chaînes de caractères issues dudit clavier (84) et lesdits moyens de lecture supplémentaires sont adressés dans une certaine cellule par des chaînes de caractères issues du clavier (84).

8. Système selon la revendication 1, caractérisé en ce que ledit moyen de traitement supplémentaire (28) inclut un moyen d'écriture supplémentaire pour la transmission d'un message de commande vers ledit régisseur logique programmable (70).

9. Système selon la revendication 8, caractérisé en ce que ledit moyen d'écriture supplémentaire inclut des chaînes de caractères désignant une fonction d'écriture de feuille de calcul et des adresses de chemin de ladite interface (24) et dudit régisseur logique programmable (70).

10. Système selon la revendication 9, caractérisé en ce que le moyen d'écriture supplémentaire inclut des chaînes de caractères désignant un certain registre dans ledit régisseur logique programmable (70) qui est destiné à recevoir ledit message de commande.

11. Système selon la revendication 1, caractérisé en ce que ledit moyen de traitement supplémentaire (28) inclut un moyen d'initialisation par menu pour déplacer l'information d'initialisation du système vers ladite interface (24).

12. Système selon la revendication 1, caractérisé en ce que ledit moyen de traitement supplémentaire (28) inclut un premier moyen fichier destiné à contenir des variables locales utilisées pour déplacer les messages directement entre lesdites cellules et ladite interface (24) et pour contenir lesdites instructions de menu, ledit moyen de traitement supplémentaire (28) incluant aussi un second moyen fichier destiné à contenir lesdites fonctions de feuille de calcul supplémentaires (96), et ledit moyen de traitement de feuille de calcul (26) incluant un moyen d'affectation destiné à affecter lesdits premier et second moyens fichiers à des emplacements de mémoire spécifiques.

13. Procédé destiné à réaliser une commande en temps réel du fonctionnement d'un équipement (42) avec un régisseur logique programmable (70), ledit régisseur logique programmable (70) pouvant recevoir des messages destiné à commander le fonctionnement dudit équipement (42) et pouvant émettre des messages indiquant un état dudit équipement (42), ledit procédé comprenant: la fourniture d'un ordinateur (22) pourvu, interconnectés de manière opérationnelle, d'un dispositif d'affichage visuel (86), d'un clavier (84), d'une mémoire (76) et d'une unité centrale de traitement, ladite fourniture incluant le fait que l'ordinateur (22) exécute des instructions de façon séquentielle, contenues dans ladite mémoire (76) en coopération avec des chaînes de caractères reçues du clavier (84); la fourniture d'une interface (24), reliée électriquement entre ledit ordinateur (22) et ledit régisseur logique programmable (70), pouvant transmettre des messages entre ledit régisseur logique programmable (70) et ledit ordinateur (22); la mise en place d'un programme de gestion de feuilles de calcul d'usage général dans ledit ordinateur (22), ledit programme de gestion de feuilles de calcul étant pourvu de cellules pouvant être affichées par ledit moyen de dispositif d'affichage visuel (86), dans lesquelles un utilisateur peut insérer une information, des fonctions de feuille de calcul qui peuvent être insérées dans lesdites feuilles de calcul par l'intermédiaire dudit clavier (84) et des instructions de menu pouvant être sélectionnées afin de réaliser des actions dans ledit programme de gestion de feuilles de calcul, ledit programme de gestion de feuilles de calcul étant normalement seulement capable de réaliser des déplacements d'information entre des fichiers de données contenus dans ladite mémoire (76) et lesdites cellules, caractérisé par la mise en place de fonctions de feuille de calcul supplémentaires (96) et d'instructions de menu supplémentaires (92) dans ledit calculateur fonctionnant par l'intermédiaire dudit programme de gestion de feuille de calcul, pour déplacer des messages directement entre lesdites cellules et ledit régisseur logique programmable (70) par l'intermédiaire de ladite interface (24), lesdits messages se déplaçant depuis lesdites cellules vers ledit régisseur logique programmable comprenant des messages de commande destinés à comman-

der le fonctionnement dudit équipement (42) et lesdits messages se déplaçant à partir dudit régisseur logique programmable (70) vers lesdites cellules comprenant lesdits messages d'état indiquant l'état dudit équipement (42) et peut demander des messages indiquant l'état dudit équipement (42) directement à partir desdites cellules et recevoir lesdits messages indiquant l'état dudit équipement (42) directement dans lesdites cellules.

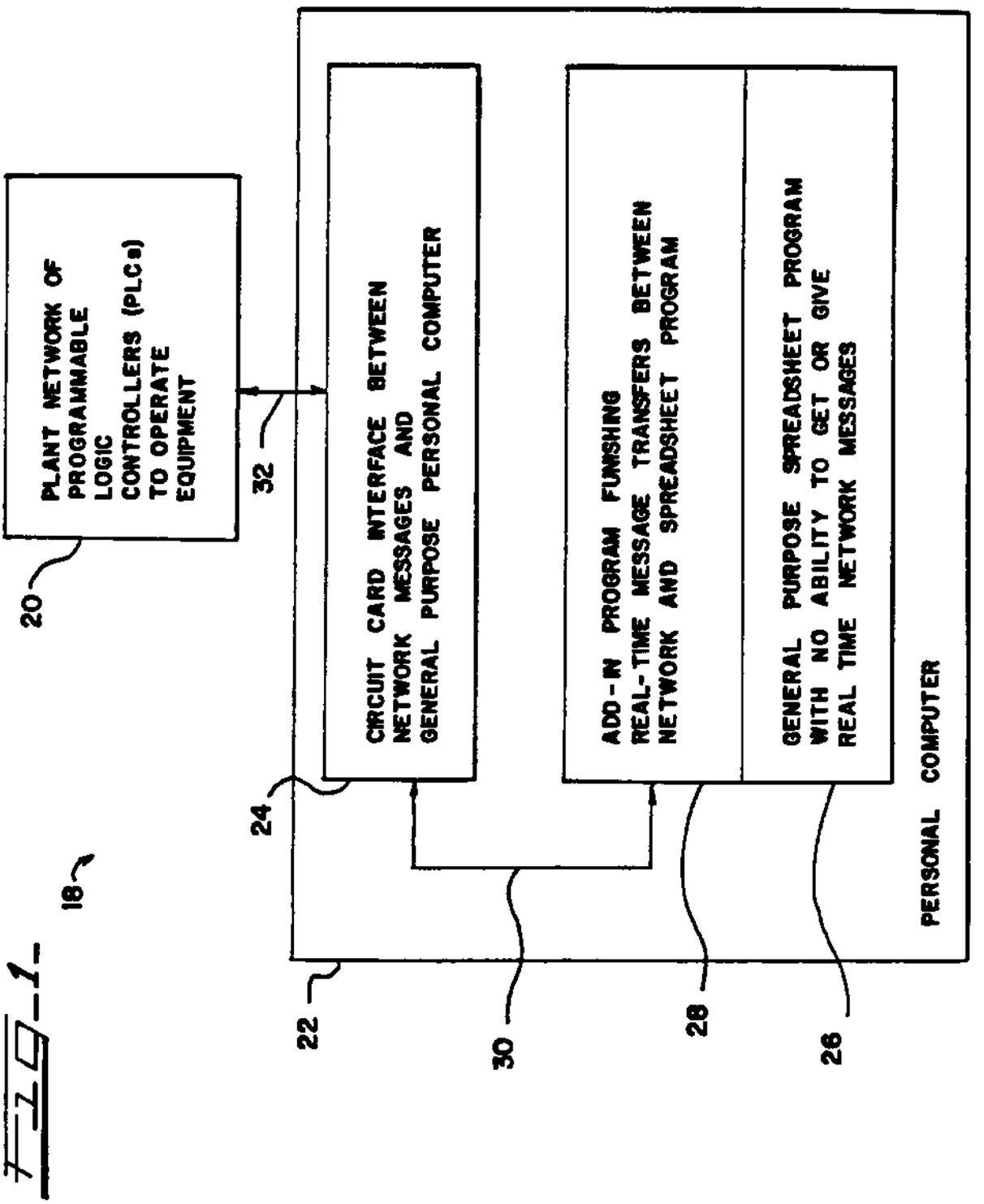

PLANT NETWORK OF PROGRAMMABLE LOGIC CONTROLLERS (PLCs) TO OPERATE EQUIPMENT
20
32
CIRCUIT CARD INTERFACE BETWEEN NETWORK MESSAGES AND GENERAL PURPOSE PERSONAL COMPUTER
24
ADD-IN PROGRAM FUNSHING REAL-TIME MESSAGE TRANSFERS BETWEEN NETWORK AND SPREADSHEET PROGRAM
GENERAL PURPOSE SPREADSHEET PROGRAM WITH NO ABILITY TO GET OR GIVE REAL TIME NETWORK MESSAGES
PERSONAL COMPUTER
18
22
30
28
26

FIG-1

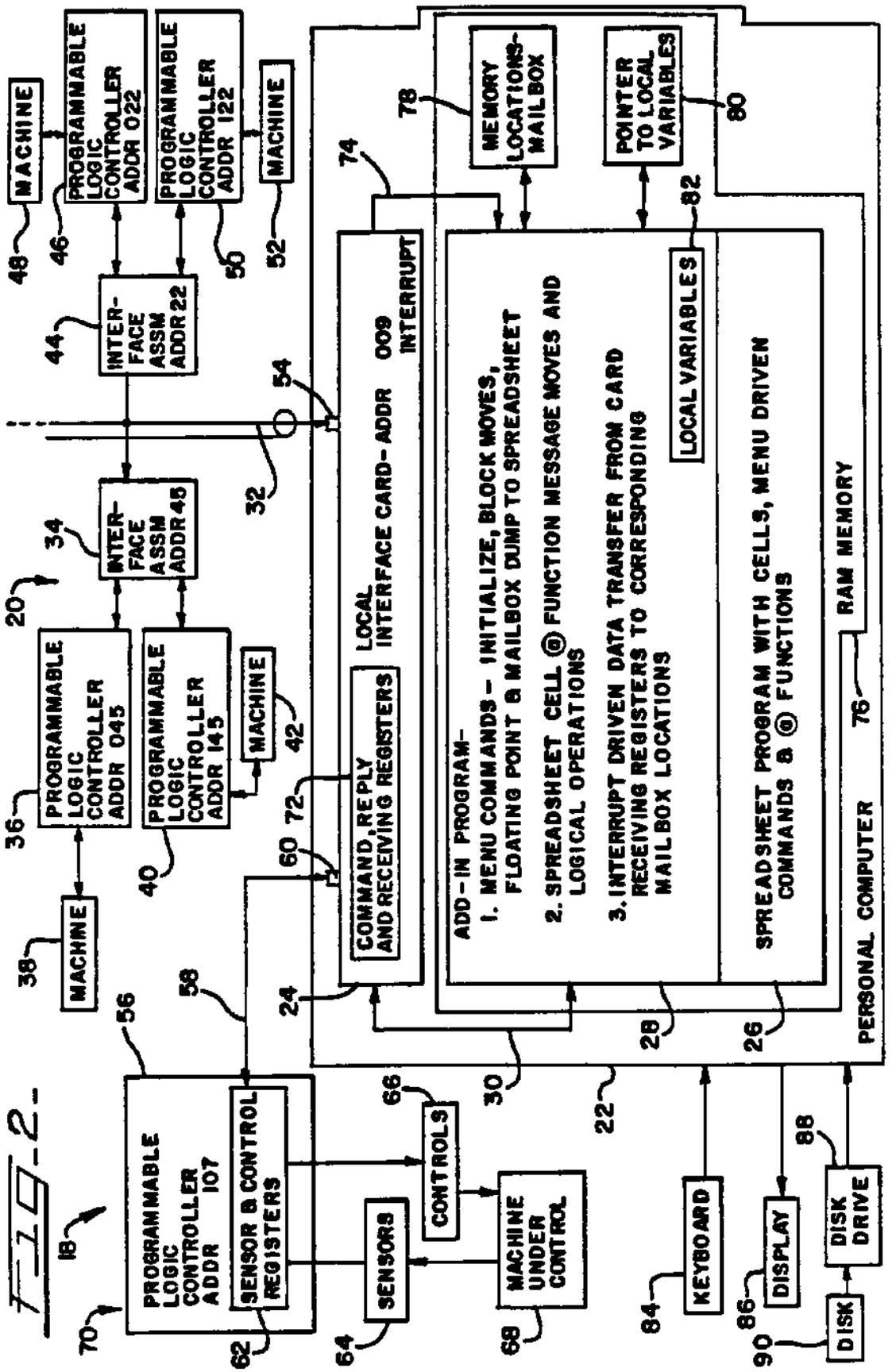

MACHINE
PROGRAMMABLE LOGIC CONTROLLER ADDR 022
PROGRAMMABLE LOGIC CONTROLLER ADDR 122
MACHINE
INTER-FACE ASSM ADDR 22
INTER-FACE ASSM ADDR 45
PROGRAMMABLE LOGIC CONTROLLER ADDR 045
PROGRAMMABLE LOGIC CONTROLLER ADDR 145
MACHINE
MACHINE
PROGRAMMABLE LOGIC CONTROLLER ADDR 107
SENSOR & CONTROL REGISTERS
SENSORS
CONTROLS
MACHINE UNDER CONTROL
LOCAL INTERFACE CARD- ADDR 009 INTERRUPT
COMMAND, REPLY AND RECEIVING REGISTERS
MEMORY LOCATIONS-MAILBOX
POINTER TO LOCAL VARIABLES
LOCAL VARIABLES
ADD-IN PROGRAM-
1. MENU COMMANDS - INITIALIZE, BLOCK MOVES, FLOATING POINT & MAILBOX DUMP TO SPREADSHEET
2. SPREADSHEET CELL @ FUNCTION MESSAGE MOVES AND LOGICAL OPERATIONS
3. INTERRUPT DRIVEN DATA TRANSFER FROM CARD RECEIVING REGISTERS TO CORRESPONDING MAILBOX LOCATIONS
SPREADSHEET PROGRAM WITH CELLS, MENU DRIVEN COMMANDS & @ FUNCTIONS
RAM MEMORY
PERSONAL COMPUTER
KEYBOARD
DISPLAY
DISK DRIVE
DISK
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90

FIG-2

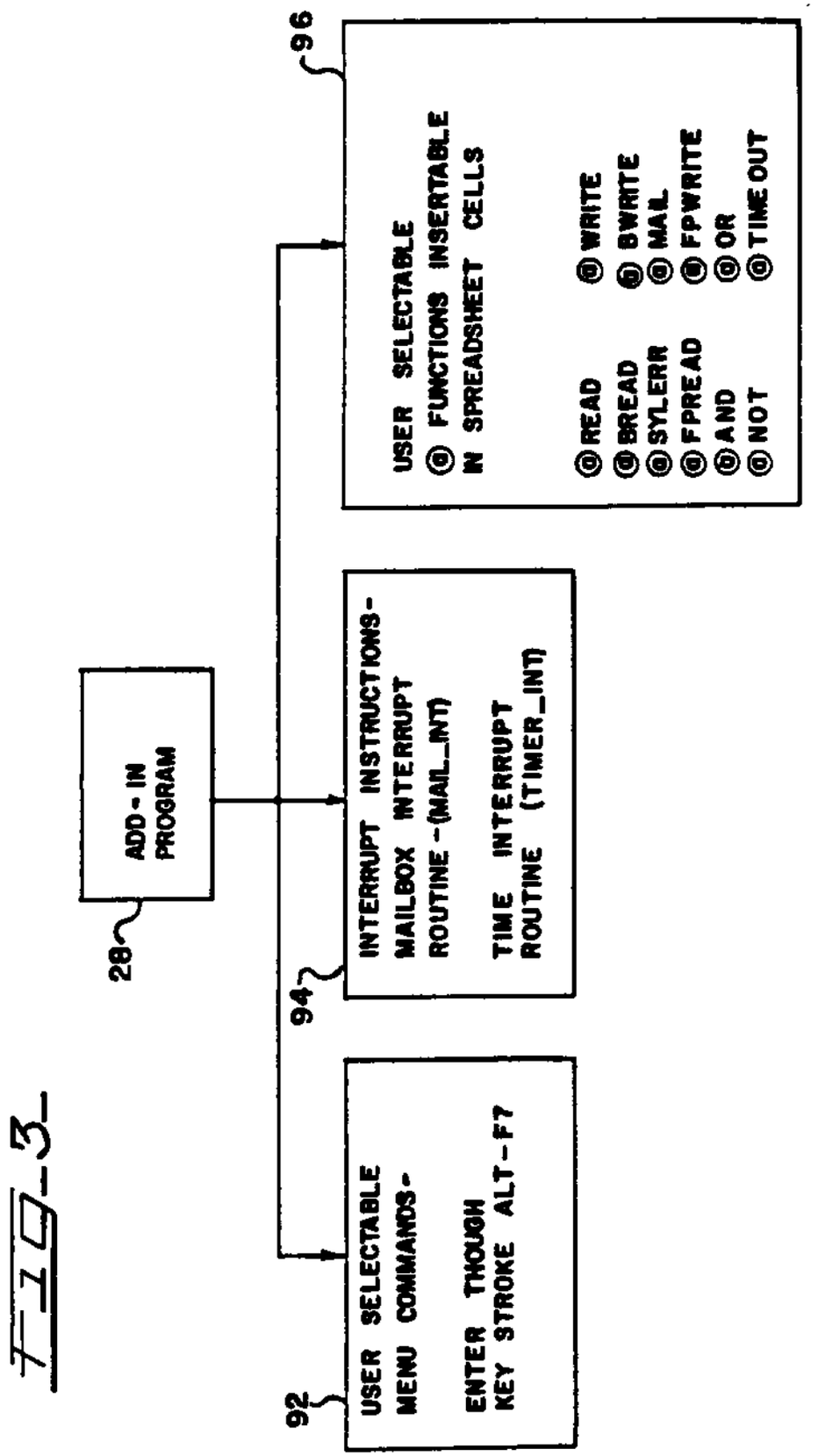
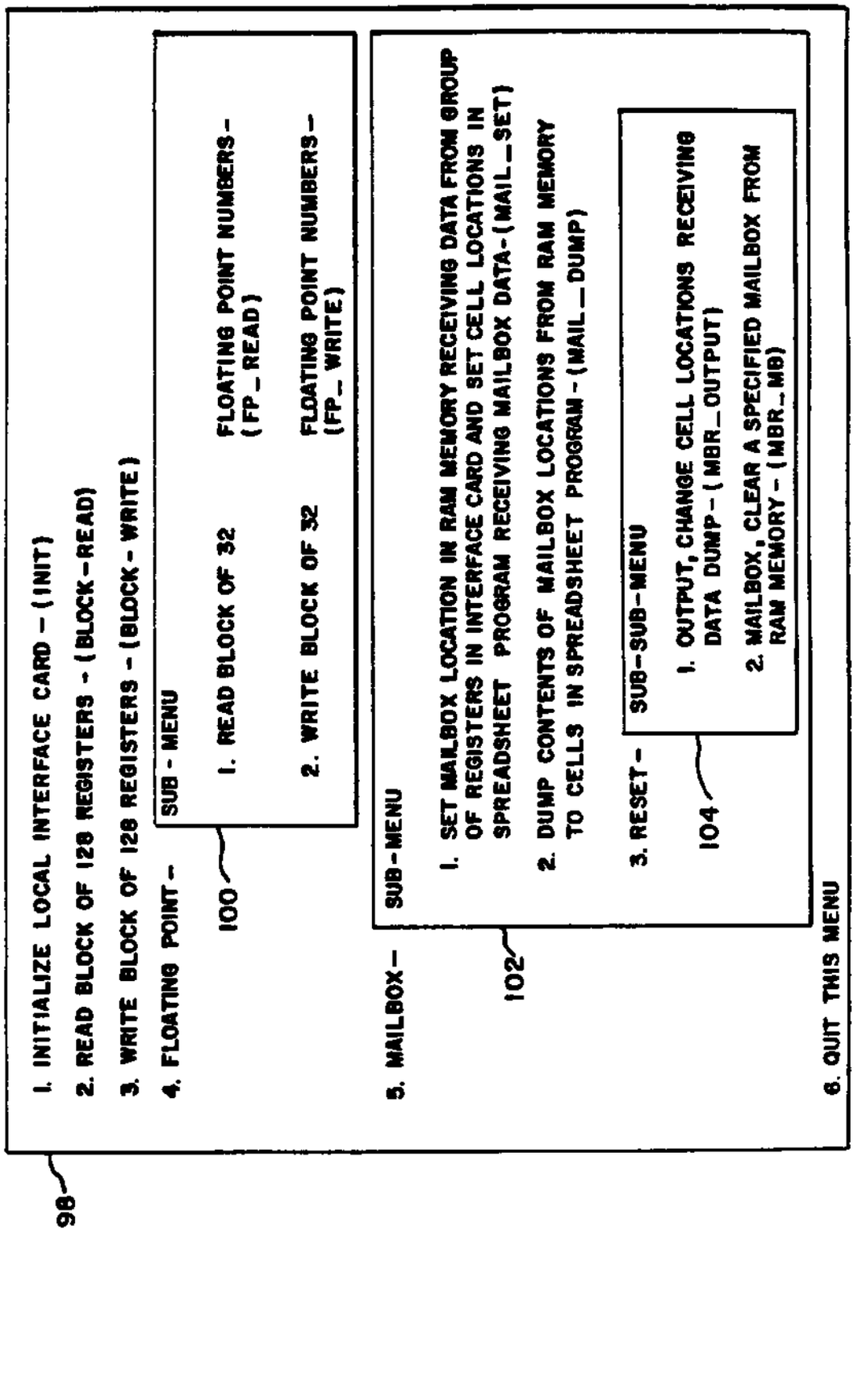

96
USER SELECTABLE
FUNCTIONS INSERTABLE
IN SPREADSHEET CELLS
READ
BREAD
SYLERR
FPREAD
AND
NOT
WRITE
BWRITE
MAIL
FPWRITE
OR
TIME OUT
ADD-IN PROGRAM
28
INTERRUPT INSTRUCTIONS-
MAILBOX INTERRUPT
ROUTINE -(MAIL_INT)
TIME INTERRUPT
ROUTINE (TIMER_INT)
94
USER SELECTABLE
MENU COMMANDS-
ENTER THOUGH
KEY STROKE ALT-F7
92

FIG-3

FIG_4


ALT - F7 MENU
1. INITIALIZE LOCAL INTERFACE CARD - (INIT)
2. READ BLOCK OF 128 REGISTERS - (BLOCK-READ)
3. WRITE BLOCK OF 128 REGISTERS - (BLOCK - WRITE)
4. FLOATING POINT -
SUB - MENU
1. READ BLOCK OF 32
FLOATING POINT NUMBERS - (FP_READ)
2. WRITE BLOCK OF 32
FLOATING POINT NUMBERS - (FP_WRITE)
5. MAILBOX -
SUB-MENU
1. SET MAILBOX LOCATION IN RAM MEMORY RECEIVING DATA FROM GROUP OF REGISTERS IN INTERFACE CARD AND SET CELL LOCATIONS IN SPREADSHEET PROGRAM RECEIVING MAILBOX DATA-(MAIL_SET)
2. DUMP CONTENTS OF MAILBOX LOCATIONS FROM RAM MEMORY TO CELLS IN SPREADSHEET PROGRAM - (MAIL_DUMP)
3. RESET -
SUB-SUB-MENU
1. OUTPUT, CHANGE CELL LOCATIONS RECEIVING DATA DUMP - (MBR_OUTPUT)
2. MAILBOX, CLEAR A SPECIFIED MAILBOX FROM RAM MEMORY - (MBR_MB)
6. QUIT THIS MENU
98
100
102
104

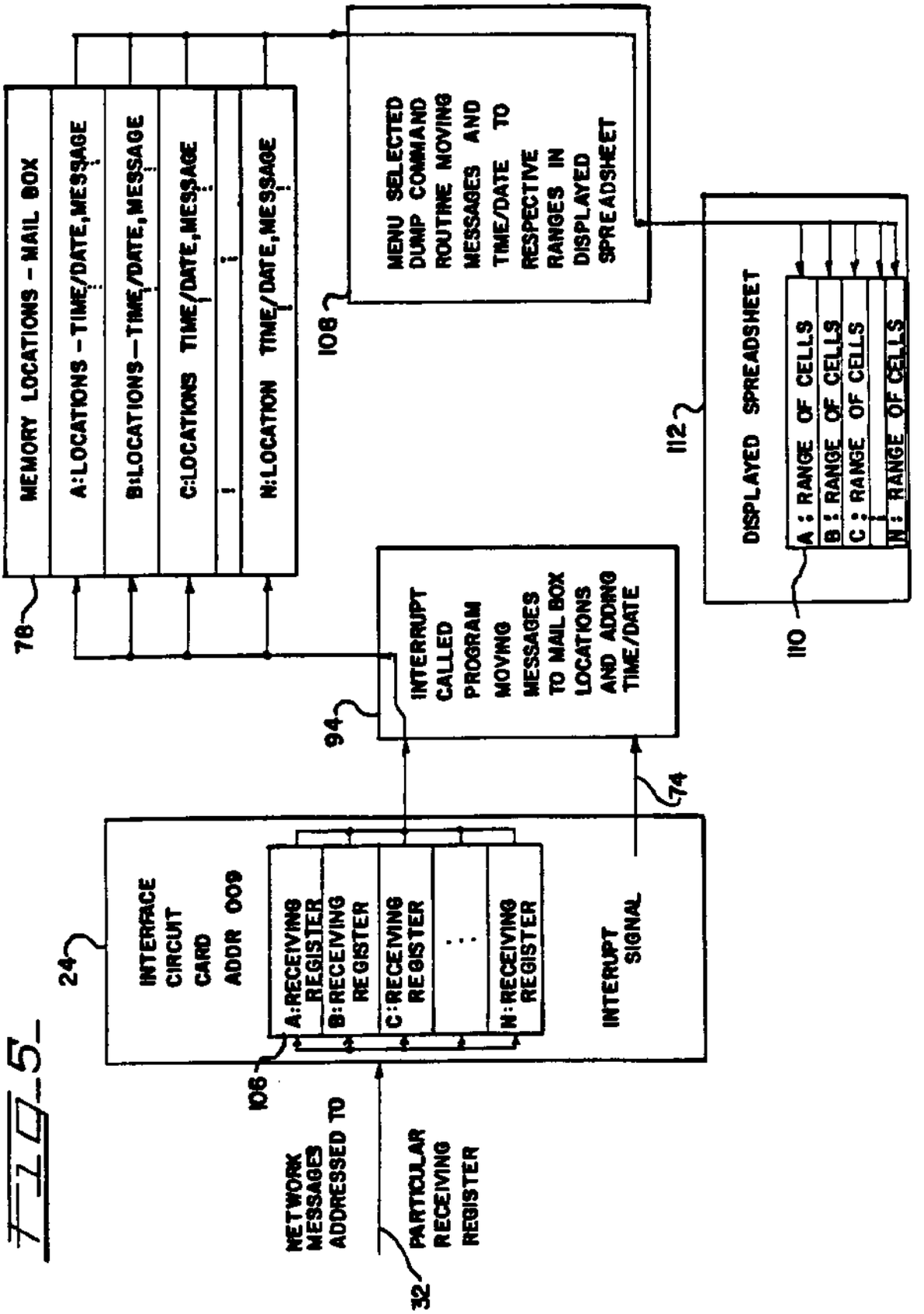

MEMORY LOCATIONS - MAIL BOX
A:LOCATIONS - TIME/DATE, MESSAGE
B:LOCATIONS - TIME/DATE, MESSAGE
C:LOCATIONS TIME/DATE, MESSAGE
N:LOCATION TIME/DATE, MESSAGE
78
MENU SELECTED DUMP COMMAND ROUTINE MOVING MESSAGES AND TIME/DATE TO RESPECTIVE RANGES IN DISPLAYED SPREADSHEET
108
DISPLAYED SPREADSHEET
A : RANGE OF CELLS
B : RANGE OF CELLS
C : RANGE OF CELLS
N : RANGE OF CELLS
112
110
INTERRUPT CALLED PROGRAM MOVING MESSAGES TO MAIL BOX LOCATIONS AND ADDING TIME/DATE
94
INTERFACE CIRCUIT CARD ADDR 009
24
A:RECEIVING REGISTER
B:RECEIVING REGISTER
C:RECEIVING REGISTER
N:RECEIVING REGISTER
INTERUPT SIGNAL
74
106
NETWORK MESSAGES ADDRESSED TO
PARTICULAR RECEIVING REGISTER
32

FIG 5

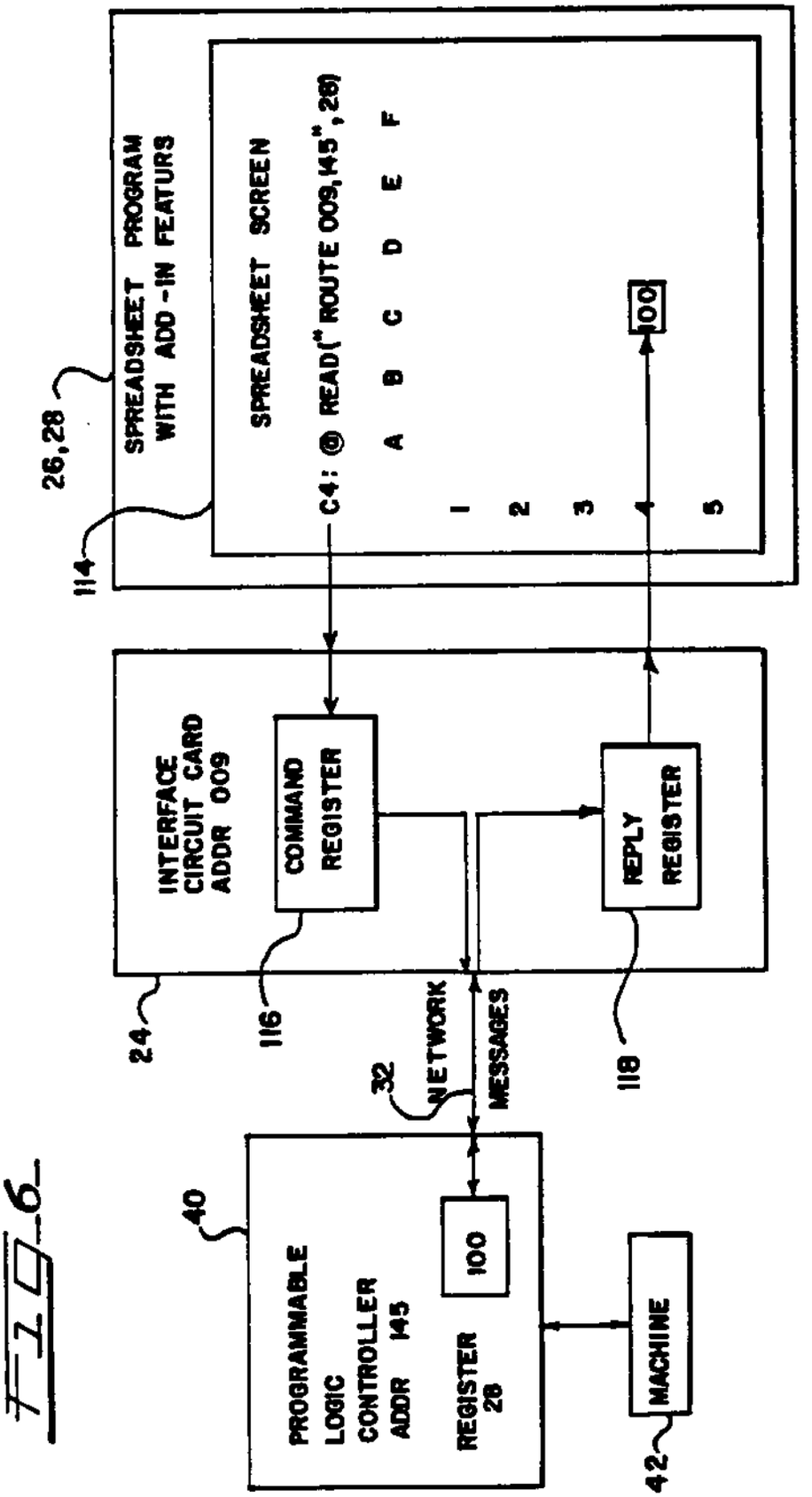

SPREADSHEET PROGRAM
WITH ADD-IN FEATURS
26,28
114
SPREADSHEET SCREEN
C4: @ READ("ROUTE 009,145",28)
A B C D E F
1
2
3
4
5
100
INTERFACE
CIRCUIT CARD
ADDR 009
24
COMMAND
REGISTER
116
REPLY
REGISTER
118
NETWORK
MESSAGES
32
PROGRAMMABLE
LOGIC
CONTROLLER
ADDR 145
40
REGISTER
28
100
MACHINE
42

FIG. 6

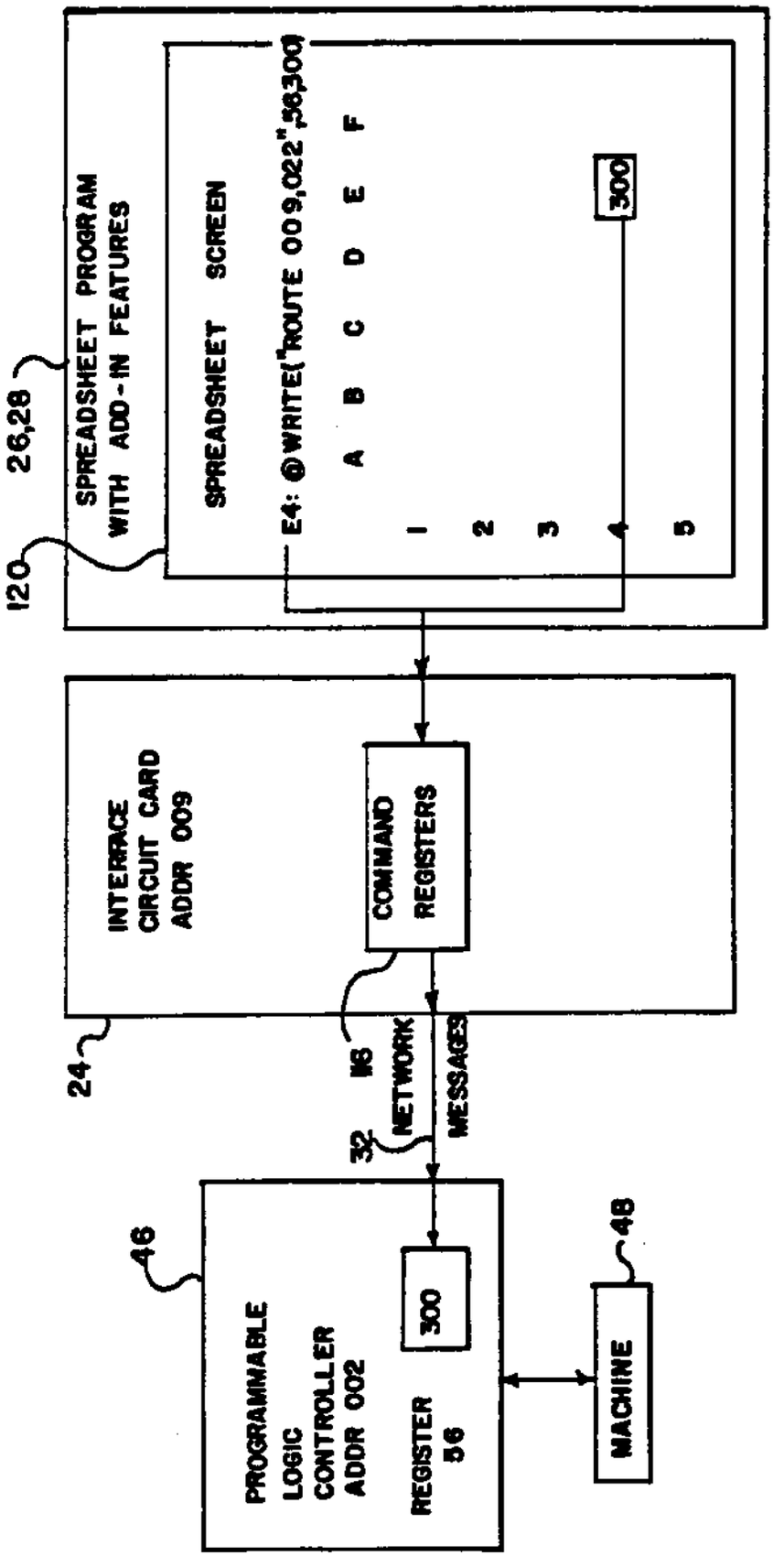

SPREADSHEET PROGRAM
WITH ADD-IN FEATURES
26,28
120
SPREADSHEET SCREEN
E4: @WRITE("ROUTE 009,022",56,300)
A B C D E F
1 2 3 4 5
300
INTERFACE
CIRCUIT CARD
ADDR 009
COMMAND
REGISTERS
24
16
NETWORK
MESSAGES
32
PROGRAMMABLE
LOGIC
CONTROLLER
ADDR 002
REGISTER
56
300
46
48
MACHINE

FIG. 7

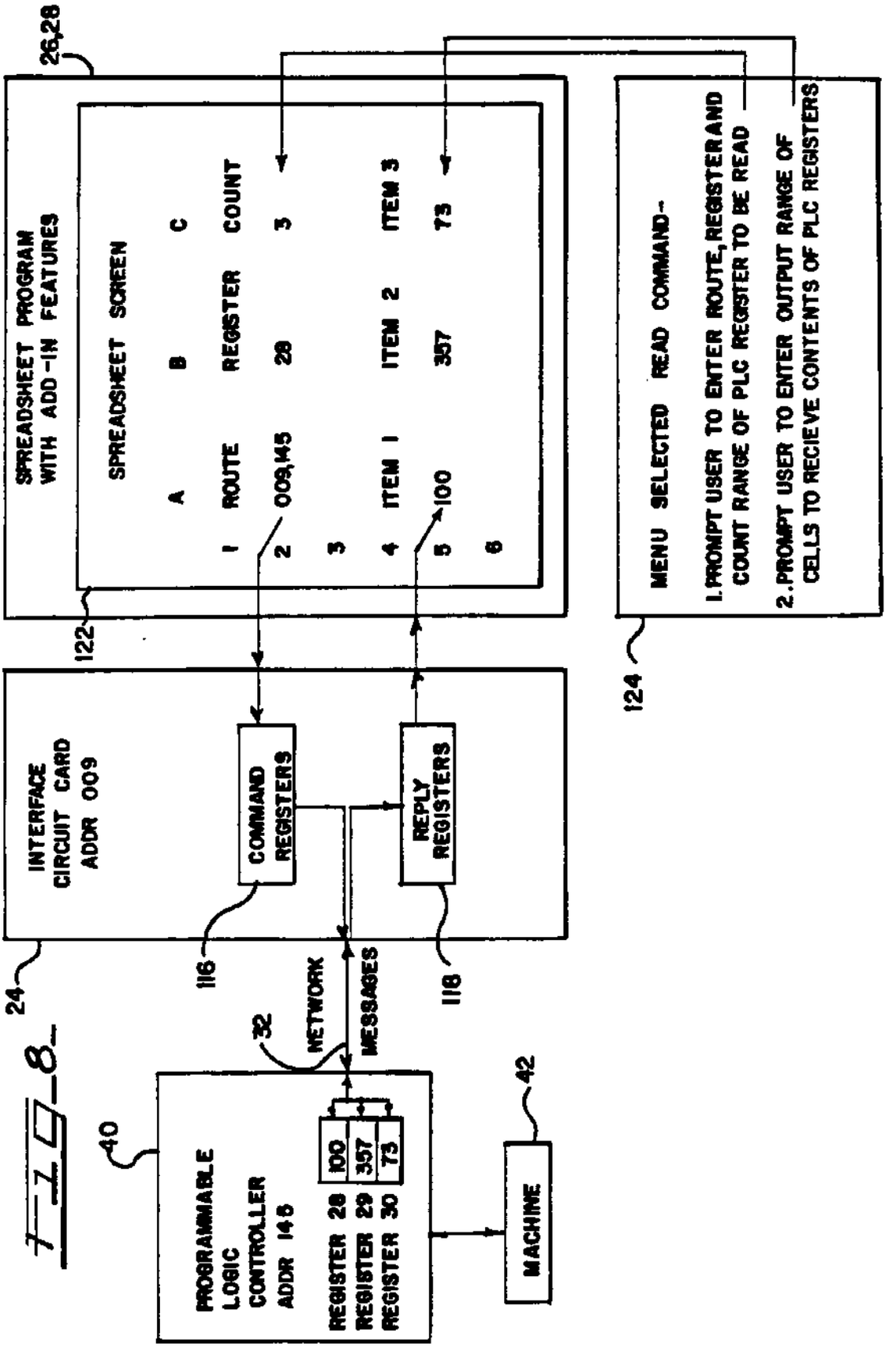

26,28
SPREADSHEET PROGRAM
WITH ADD-IN FEATURES
SPREADSHEET SCREEN
A B C
1 ROUTE REGISTER COUNT
2 009,145 28 3
3
4 ITEM 1 ITEM 2 ITEM 3
5 100 357 73
6
122
MENU SELECTED READ COMMAND-
1. PROMPT USER TO ENTER ROUTE, REGISTER AND COUNT RANGE OF PLC REGISTER TO BE READ
2. PROMPT USER TO ENTER OUTPUT RANGE OF CELLS TO RECIEVE CONTENTS OF PLC REGISTERS
124
INTERFACE CIRCUIT CARD ADDR 009
COMMAND REGISTERS
REPLY REGISTERS
116
118
24
NETWORK
MESSAGES
32
PROGRAMMABLE LOGIC CONTROLLER ADDR 145
REGISTER 28
REGISTER 29
REGISTER 30
100
357
73
40
MACHINE
42

FIG 8

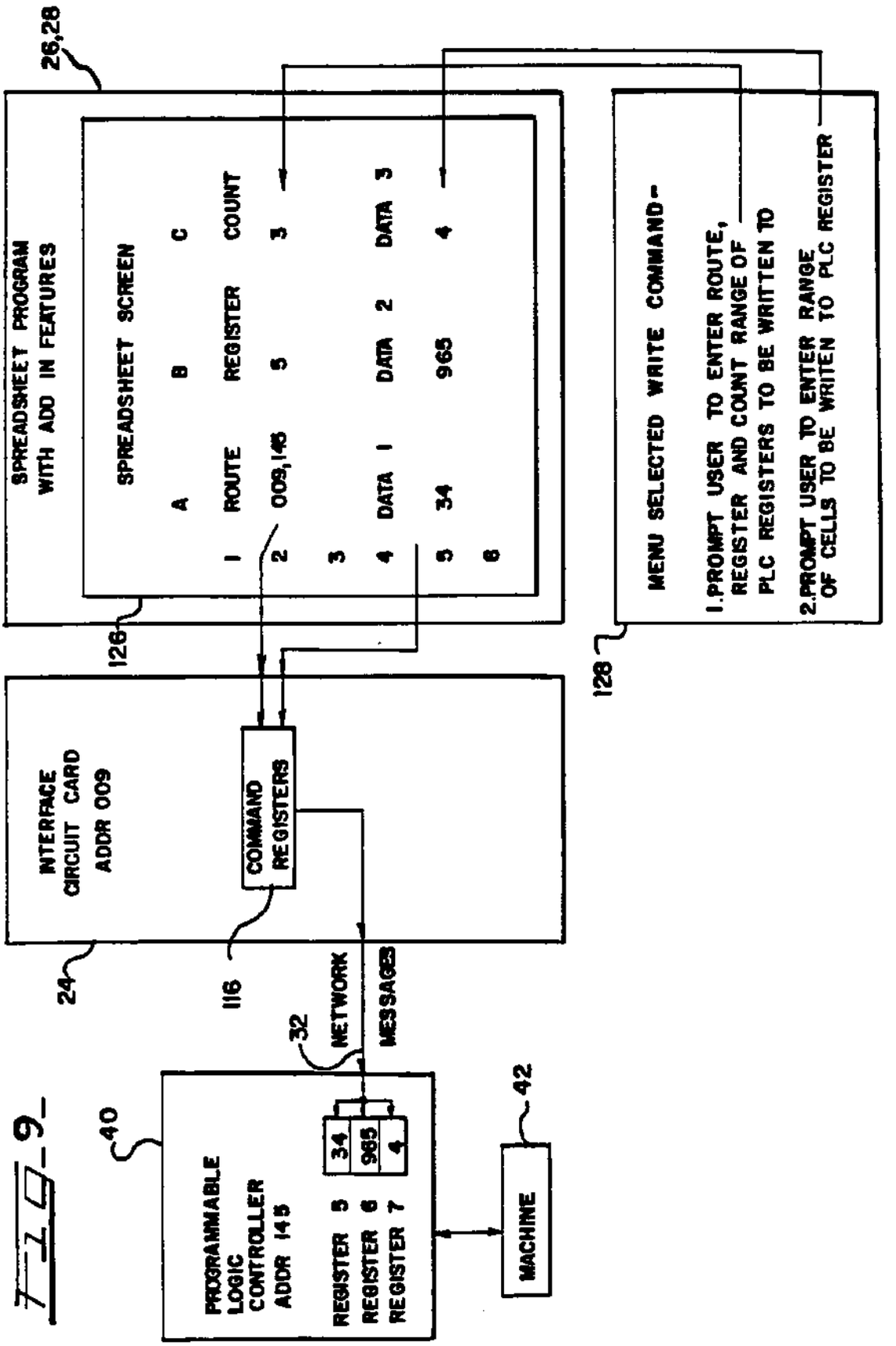

26,28
SPREADSHEET PROGRAM
WITH ADD IN FEATURES
SPREADSHEET SCREEN
A B C
1 ROUTE REGISTER COUNT
2 009,145 5 3
3
4 DATA 1 DATA 2 DATA 3
5 34 965 4
6
126
MENU SELECTED WRITE COMMAND-
1.PROMPT USER TO ENTER ROUTE,
REGISTER AND COUNT RANGE OF
PLC REGISTERS TO BE WRITTEN TO
2.PROMPT USER TO ENTER RANGE
OF CELLS TO BE WRITEN TO PLC REGISTER
128
INTERFACE
CIRCUIT CARD
ADDR 009
COMMAND
REGISTERS
24
116
32
NETWORK
MESSAGES
40
PROGRAMMABLE
LOGIC
CONTROLLER
ADDR 145
REGISTER 5
REGISTER 6
REGISTER 7
34
965
4
42
MACHINE

FIG-9

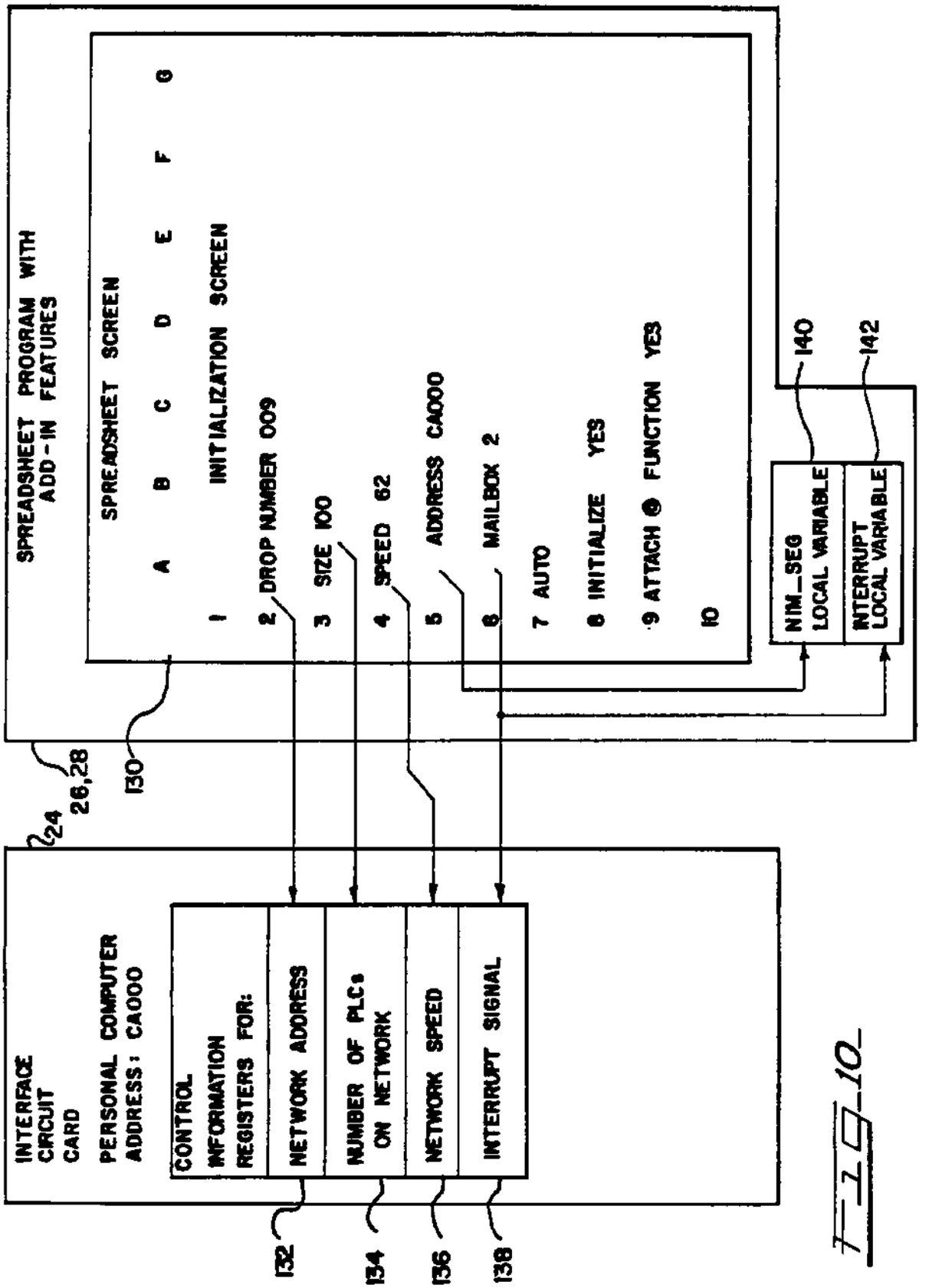

SPREADSHEET PROGRAM WITH
ADD-IN FEATURES
SPREADSHEET SCREEN
A B C D E F G
1 INITIALIZATION SCREEN
2 DROP NUMBER 009
3 SIZE 100
4 SPEED 62
5 ADDRESS CA000
6 MAILBOX 2
7 AUTO
8 INITIALIZE YES
9 ATTACH # FUNCTION YES
10
NIM_SEG
LOCAL VARIABLE
INTERRUPT
LOCAL VARIABLE
140
142
130
26,28
24
INTERFACE
CIRCUIT
CARD
PERSONAL COMPUTER
ADDRESS: CA000
CONTROL
INFORMATION
REGISTERS FOR:
NETWORK ADDRESS
NUMBER OF PLCs
ON NETWORK
NETWORK SPEED
INTERRUPT SIGNAL
132
134
136
138

FIG_10

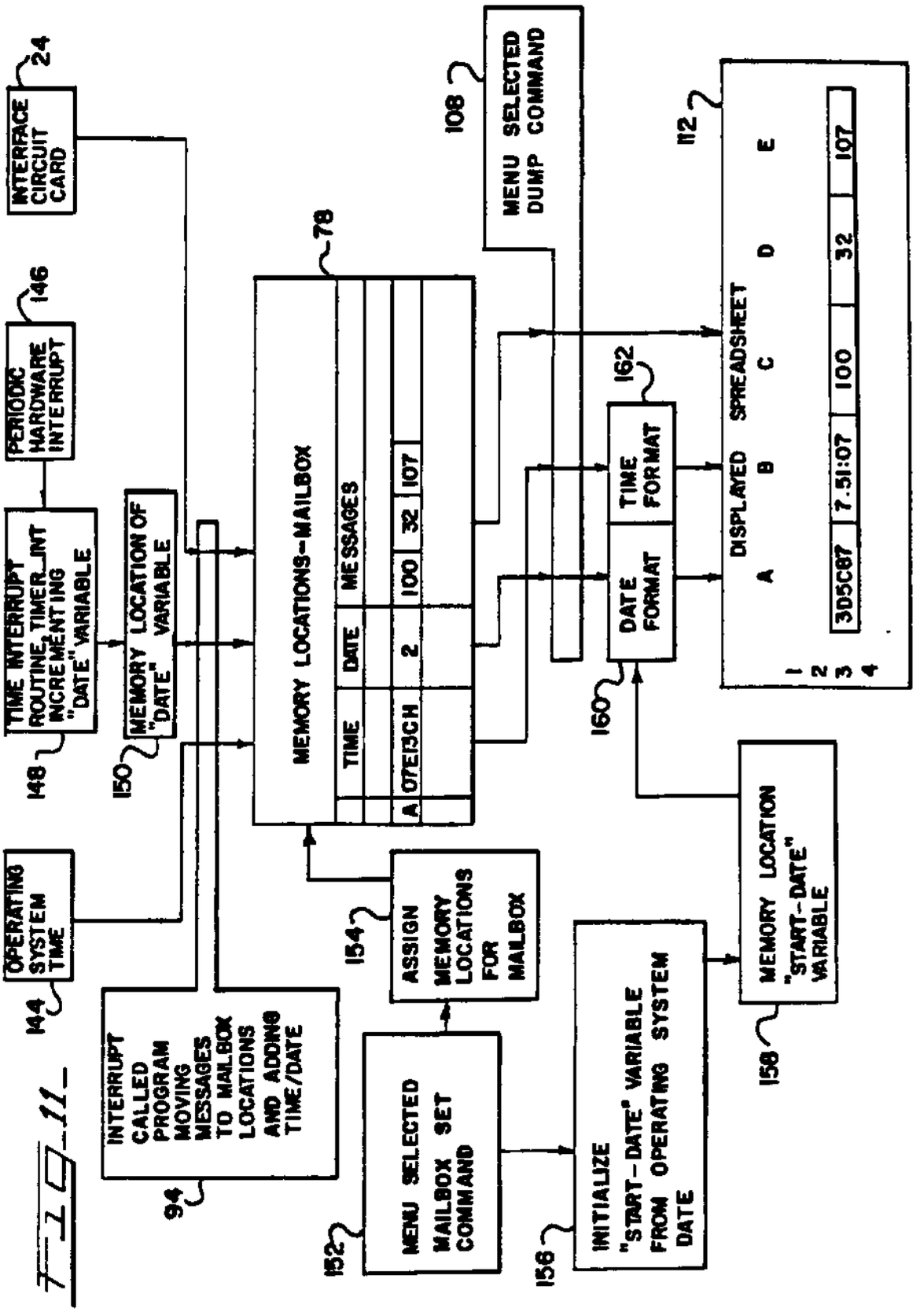

INTERFACE CIRCUIT CARD
24
PERIODIC HARDWARE INTERRUPT
146
TIME INTERRUPT ROUTINE, TIMER_INT INCREMENTING "DATE" VARIABLE
148
MEMORY LOCATION OF "DATE" VARIABLE
150
OPERATING SYSTEM TIME
144
INTERRUPT CALLED PROGRAM MOVING MESSAGES TO MAILBOX LOCATIONS AND ADDING TIME/DATE
94
MEMORY LOCATIONS-MAILBOX
78
TIME
DATE
MESSAGES
A
07E13CH
2
100
32
107
MENU SELECTED DUMP COMMAND
108
DISPLAYED SPREADSHEET
112
A
B
C
D
E
1
2
3
4
3D5C87
7:51:07
100
32
107
TIME FORMAT
162
DATE FORMAT
160
ASSIGN MEMORY LOCATIONS FOR MAILBOX
154
MENU SELECTED MAILBOX SET COMMAND
152
INITIALIZE "START-DATE" VARIABLE FROM OPERATING SYSTEM DATE
156
MEMORY LOCATION "START-DATE" VARIABLE
158

FIG_11_

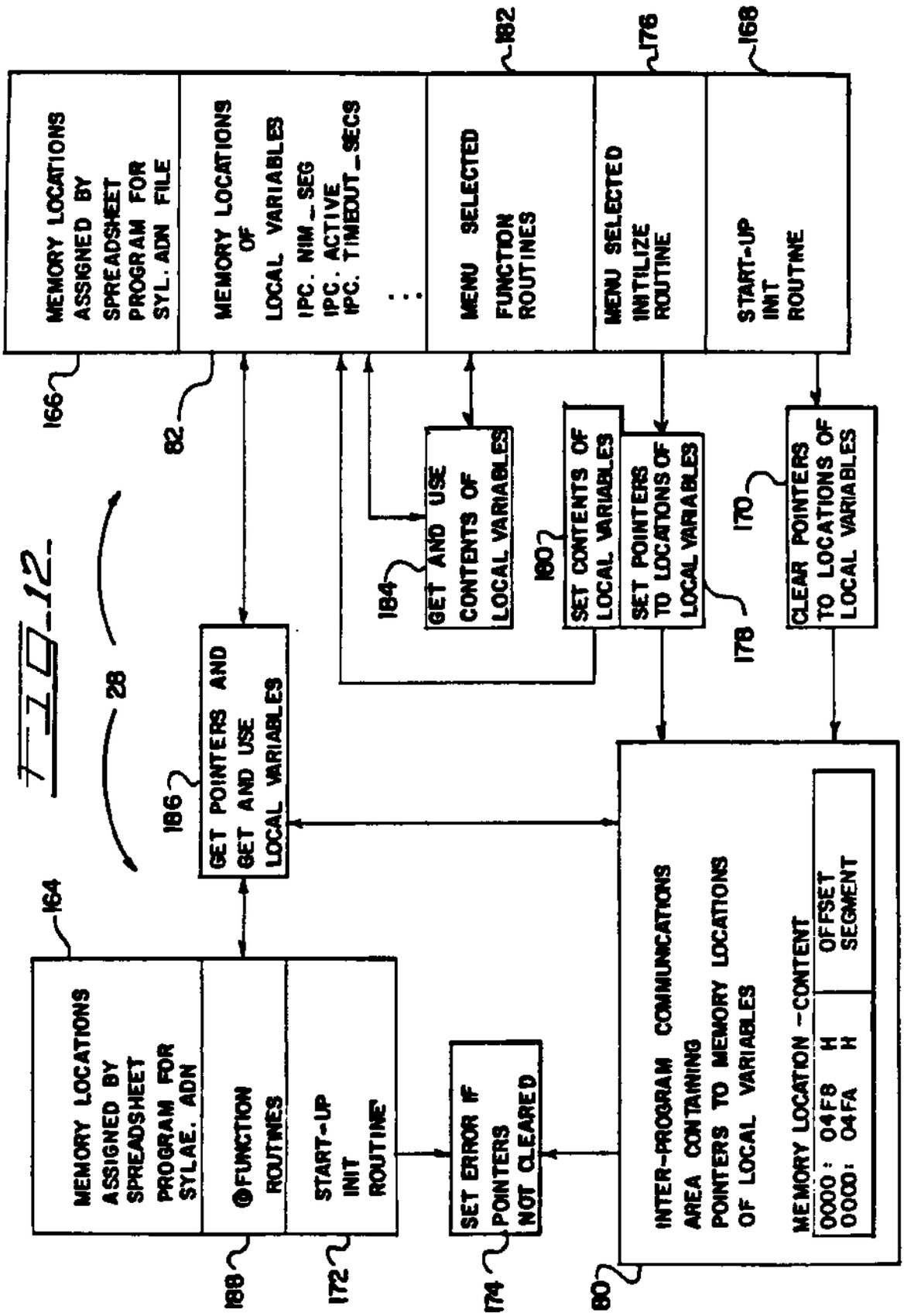

MEMORY LOCATIONS ASSIGNED BY SPREADSHEET PROGRAM FOR SYL. ADN FILE
MEMORY LOCATIONS OF LOCAL VARIABLES
IPC. NIM_SEG
IPC. ACTIVE
IPC. TIMEOUT_SECS
MENU SELECTED FUNCTION ROUTINES
MENU SELECTED INITILIZE ROUTINE
START-UP INIT ROUTINE
166
82
182
176
168
GET AND USE CONTENTS OF LOCAL VARIABLES
184
SET CONTENTS OF LOCAL VARIABLES
180
SET POINTERS TO LOCATIONS OF LOCAL VARIABLES
178
CLEAR POINTERS TO LOCATIONS OF LOCAL VARIABLES
170
GET POINTERS AND GET AND USE LOCAL VARIABLES
186
28
FIG-12
MEMORY LOCATIONS ASSIGNED BY SPREADSHEET PROGRAM FOR SYLAE. ADN
FUNCTION ROUTINES
START-UP INIT ROUTINE
164
188
172
SET ERROR IF POINTERS NOT CLEARED
174
INTER-PROGRAM COMMUNICATIONS AREA CONTAINING POINTERS TO MEMORY LOCATIONS OF LOCAL VARIABLES
MEMORY LOCATION -CONTENT
0000: 04F8 H
0000: 04FA H
OFFSET
SEGMENT
80

FIG_13_

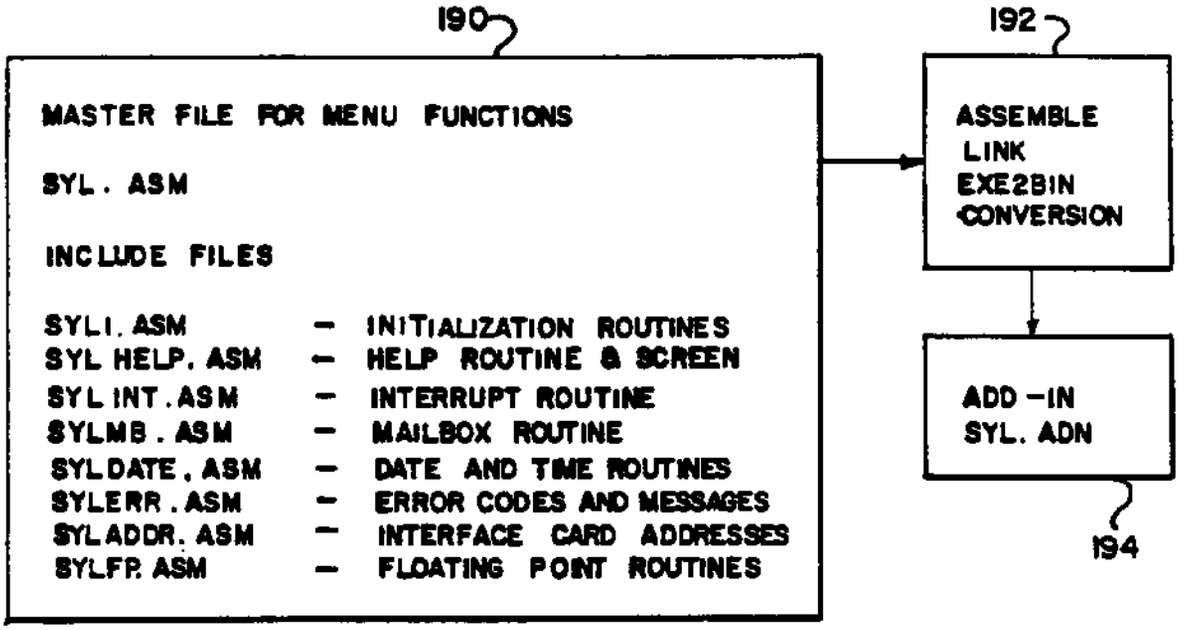
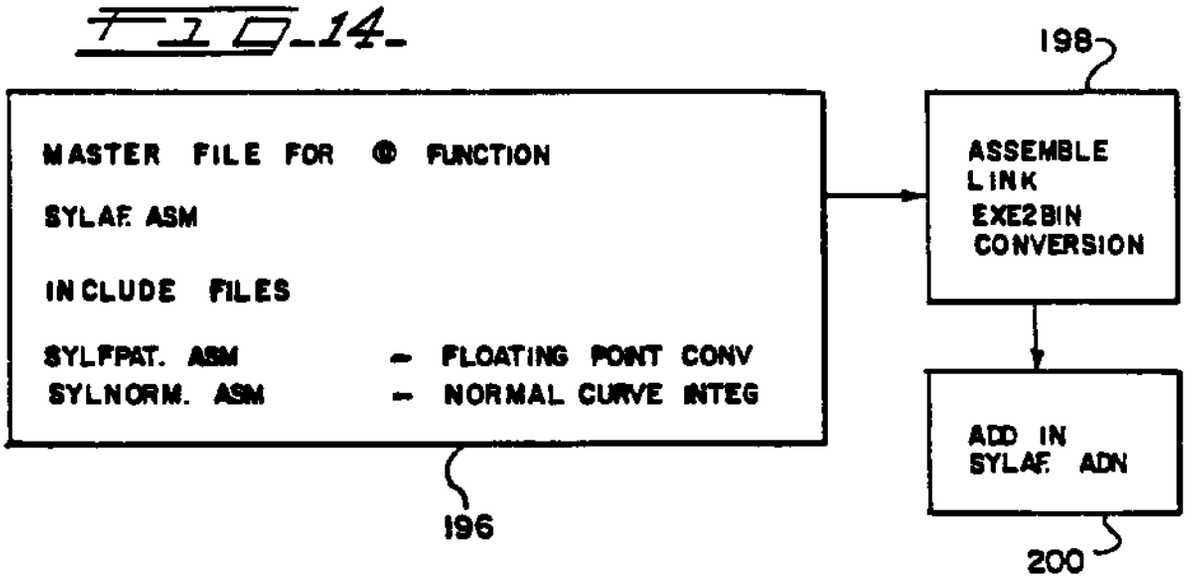

190
MASTER FILE FOR MENU FUNCTIONS
SYL. ASM
INCLUDE FILES
SYLI. ASM – INITIALIZATION ROUTINES
SYL HELP. ASM – HELP ROUTINE & SCREEN
SYL INT. ASM – INTERRUPT ROUTINE
SYLMB. ASM – MAILBOX ROUTINE
SYLDATE. ASM – DATE AND TIME ROUTINES
SYLERR. ASM – ERROR CODES AND MESSAGES
SYLADDR. ASM – INTERFACE CARD ADDRESSES
SYLFP. ASM – FLOATING POINT ROUTINES
192
ASSEMBLE
LINK
EXE2BIN
CONVERSION
ADD –IN
SYL. ADN
194

FIG_14_


MASTER FILE FOR @ FUNCTION
SYLAF. ASM
INCLUDE FILES
SYLFPAT. ASM – FLOATING POINT CONV
SYLNORM. ASM – NORMAL CURVE INTEG
196
198
ASSEMBLE
LINK
EXE2BIN
CONVERSION
ADD IN
SYLAF. ADN
200