

NAT'L INST. OF STAND & TECH R.I.C.



A11104 232706

NATIONAL INSTITUTE OF STANDARDS &
TECHNOLOGY
Research Information Center
Gaithersburg, MD 20899

NBSIR 88-3769

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

NEW NBS PUB
JUL 14 1988

April 21, 1988

By:
C. Denver Lovett



AMRF

**THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS
OF
THE AUTOMATED MANUFACTURING RESEARCH FACILITY
AT THE NATIONAL BUREAU OF STANDARDS**

A Report
Prepared by C. Denver Lovett
Date: April 21, 1988

Certain commercial equipment and software are identified in this paper in order to adequately specify the experimental facility. Such identification does not imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that the equipment identified is necessarily the best available for the purpose.

This publication was prepared by United States Government employees as part of their official duties and is, therefore, a work of the U.S. Government and is not subject to copyright.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

CONTENTS

	Page
1.0 INTRODUCTION.....	1
1.1 PURPOSE.....	1
1.2 Intended Audience.....	1
1.3 Overview.....	1
2.0 CENTRAL EQUIPMENT CONTROLLER (CEC).....	5
2.1 COMPUTER SYSTEM ORGANIZATION.....	5
2.2 Software Structure.....	5
2.3 Software Operating Features.....	6
2.3.1 Communication With Workstation.....	8
2.3.2 Structure of the Command String.....	8
2.3.3 Parsing Commands.....	13
2.3.4 Scheduling Event.....	13
2.3.5 Designating and Managing Mailboxes.....	15
2.3.6 Obtaining Statuses.....	15
2.3.7 VWS Rules for Issuing Commands.....	16
2.3.8 Software Safety Interlocks.....	19
3.0 GE 2000 CONTROLLER.....	21
3.1 DNC COMMUNICATION INTERFACE.....	22
3.2 Communication Protocol.....	24
3.2.1 Command Message.....	24
3.2.2 Part Program Download in Level 2 Protocol.....	26
3.2.3 Part Program Upload in Level 2 Protocol.....	26
3.2.4 Select Program for Execution.....	27
3.2.5 Delete Program from GE-2000 Memory.....	27
3.2.6 Retrieve Tool Length Offset Table.....	28
3.2.7 Machine Tool Status.....	28
4.0 THE UNIMATION VAL II CONTROLLER.....	29
4.1 INTRODUCTION: THE CEC ROBOT PROGRAM.....	29
4.2 Programming Modes for VAL II.....	30
4.3 VAL II Supervisory System.....	31
4.3.1 VAL II Supervisory Protocol.....	32
4.3.2 DDCMP Protocol.....	33
4.3.3 VAL II Message Protocol.....	38
4.3.4 Message Format.....	41
4.4 Robot Communication Software.....	41
5.0 HYDRAULIC CONTROLLER.....	49
5.1 MAJOR SOFTWARE FEATURES.....	52
6.0 CONCLUDING OBSERVATIONS.....	52
APPENDIX I. EQUIPMENT CONTROLLER'S COMMAND SETS.....	53
REFERENCES.....	58

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

LIST OF FIGURES

	Page
Figure 1. CEC Programs and Other Controllers.....	3
Figure 2. Operation of VWS_Control Program.....	7
Figure 3. Diagram Showing Relationship Between..... SUN_COMM and its Subroutines	9
Figure 4. A Scheme for Activating Programs.....	14
Figure 5. Interaction of SUN_COMM and VWS_STATUS Program.....	17
Figure 6. Leighton Diagram for Monarch Program.....	18
Figure 7. A Diagram Showing Relationship Between VWS_STATUS..... and its Subroutines	23
Figure 8. Format for Numbered Data Message.....	35
Figure 9. General Format for Unnumbered Control Message.....	36
Figure 10. VAL Data Message Format.....	40
Figure 11. Format of Transmitted Message.....	40
Figure 12. Leighton Diagram for Robot Program.....	42
Figure 13. Flowchart for Robot Move from Location "Safe"..... to Work Volume Locations and Return to "Safe"	44
Figure 14. Flowchart for Subroutine Robot:..... (Part A LU2 Routine)	45
Figure 15. Flowchart for Subroutine Robot:..... (Part B of LU2 Routine)	46
Figure 16. Flowchart for Subroutine Robot:..... (Part C of LU2 Routine)	47
Figure 17. Flowchart for Subroutine Robot:..... (Part D of LU2 Routine)	48
Figure 18. ENQ Routine of Subroutine Norm_val_msg..... for Retrieving and Monitoring Control Messages	50
Figure 19. SOH Routine of Subroutine Norm_val_msg..... for Retrieving Data Messages	51

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

LIST OF TABLES

	Page
Table 1. Typical Workstation Operation.....	12
Table 2. Equipment Status Commands.....	19
Table 3. Software Interlock Commands.....	21
Table 4. Types of DDCMP Control Messages.....	37
Table 5. Control Message and Data Message Sequence.....	39

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

ABSTRACT

This document describes how the central equipment controller (CEC) of the Vertical Workstation (VWS) operates as an interfacing controller between an upper-level controller, the workstation controller (WC), and three lower-level controllers and two data acquisition units. The CEC usually acts as a master controller when communicating with the three lower-level controllers and as a slave controller when communicating with the WC. The three lower-level controllers are computer-based. One of these is a robot controller, one is a machine tool controller and the third is a general purpose computer operating as a hydraulic controller. Each controller operates under a different communication protocol. The robot controller operates in a monitor command mode, the machine tool controller operates in a DNC mode (with NC program transfers) and the hydraulic controller operates in a command-driven program mode. The sensor inputs for achieving hardware handshaking are retrieved from the two data acquisition controllers. At the heart of the CEC is a collection of eight programs that are used to coordinate the activities of the three lower-level controllers and devices such as the vacuum and the NBS-designed robot gripper. The CEC operates along with the WC, in a multi-controller and multi-programming environment, to achieve a high degree of machining flexibility by employing VWS features such as: (1) an automated NC-code generator, (2) an automated process planning system and (3) an automated robot programming generator. These features make the VWS a highly integrated and a highly automated machining system.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS OF
THE AUTOMATED MANUFACTURING RESEARCH FACILITY
AT THE NATIONAL BUREAU OF STANDARDS

1.0 INTRODUCTION

The Automated Manufacturing Facility (AMRF) at the National Bureau of standards (NBS) consists of three machining workstations, a deburring workstation, an inspection workstation and a material handling system, all acting under the coordination of a cell controller which is data-driven. A major feature of these data-driven workstations is that the actions of the machine tools and robots are determined by a computerized description of the part to be manufactured. One of the three machining workstations of the AMRF is the Vertical Workstation (VWS). This document deals exclusively with those VWS equipment controllers that are located on the shop floor of the AMRF facility.

1.1 PURPOSE

The purpose of this document is to explain how the equipment controllers of the Vertical Workstation (VWS) operate together to perform the main activities of the workstation. These controllers have been programmed to: (1) execute commands received from the workstation controller, (2) determine statuses of equipment, (3) transfer programs between controllers and (4) resolve conflicts between equipment competing for the same resources. The coordination of these controllers can best be explained by developing a framework for understanding the task allocations among these controllers and explaining how statuses of workstation equipment are assessed. The major intent of this document is to describe the software communication interfaces between controllers and to describe how the eight command-driven programs operate in allocating tasks among controllers. Although the workstation controller is mentioned throughout this document, it is not the purpose of this document to describe the workstation's controller. It is discussed elsewhere [KRA1-3] [K&J1] [NA&J].

1.2 INTENDED AUDIENCE

This document is intended for those readers who would develop communication interfaces and other computer programs to enable different types of equipment controllers to work together in a coordinated manner to perform the activities of an automated manufacturing system. This document may also be useful to those who are interested in obtaining a general understanding of the architecture of the VWS.

1.3 OVERVIEW

The VWS consists of:

- (a) a Sun workstation controller (WC) that provides overall direction of the workstation activities,

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

- (b) a Monarch vertical machining center (VMC 75),
- (c) a Unimate 4070 Robot
- (d) a VAL II controller that supervises the activities of the Robot,
- (e) a central equipment controller which is an HP-9000, series 520 computer,
- (f) a hydraulic controller which is an HP-9836, series 200 computer,
- (g) a GE-2000 controller that directs the overall activities of the vertical machining center (VMC),
- (h) an automated vise and an automated pallet on the table of the VMC to fixture the workpiece,
- (i) a number of sensors and actuators,
- (j) two data acquisition units to activate devices and obtain the statuses of devices and equipment and,
- (k) two part trays and two roller tables to receive material and delivery parts.

The Sun WC resides in a control room of the AMRF. All the other equipment is housed on the shop floor.

The VWS is a highly integrated and automated machining system. Its main goal is to achieve a high degree of machining flexibility by employing a computer-aided design system and computer-aided manufacturing system. Important features of the VWS are: (1) an automated NC-code generator that is driven on-line by a computer-aided design system, (2) an automated process planning system, (3) an automated robot-program generator and (4) the ability to run either in the stand-alone mode or subordinate to the AMRF cell. All of these features are implemented at the workstation controller [KRA1-3, K&J1, NA&J]. The workstation controller downloads the complete NC-program to the CEC, and in turn the CEC downloads the NC program to the GE-2000 controller. This completed NC program contains NC-code that also carries out the process plans that were generated by the automated process planing system.

Figure 1 shows the relationship between the WC, the CEC, the other two controllers of the VWS and the two data acquisition units. The CEC communicates with the WC via a RS-232 interface. It communicates with the robot controller and the machine tool controller via RS-232 interfaces and it communicates with the hydraulic controller and one of the two data

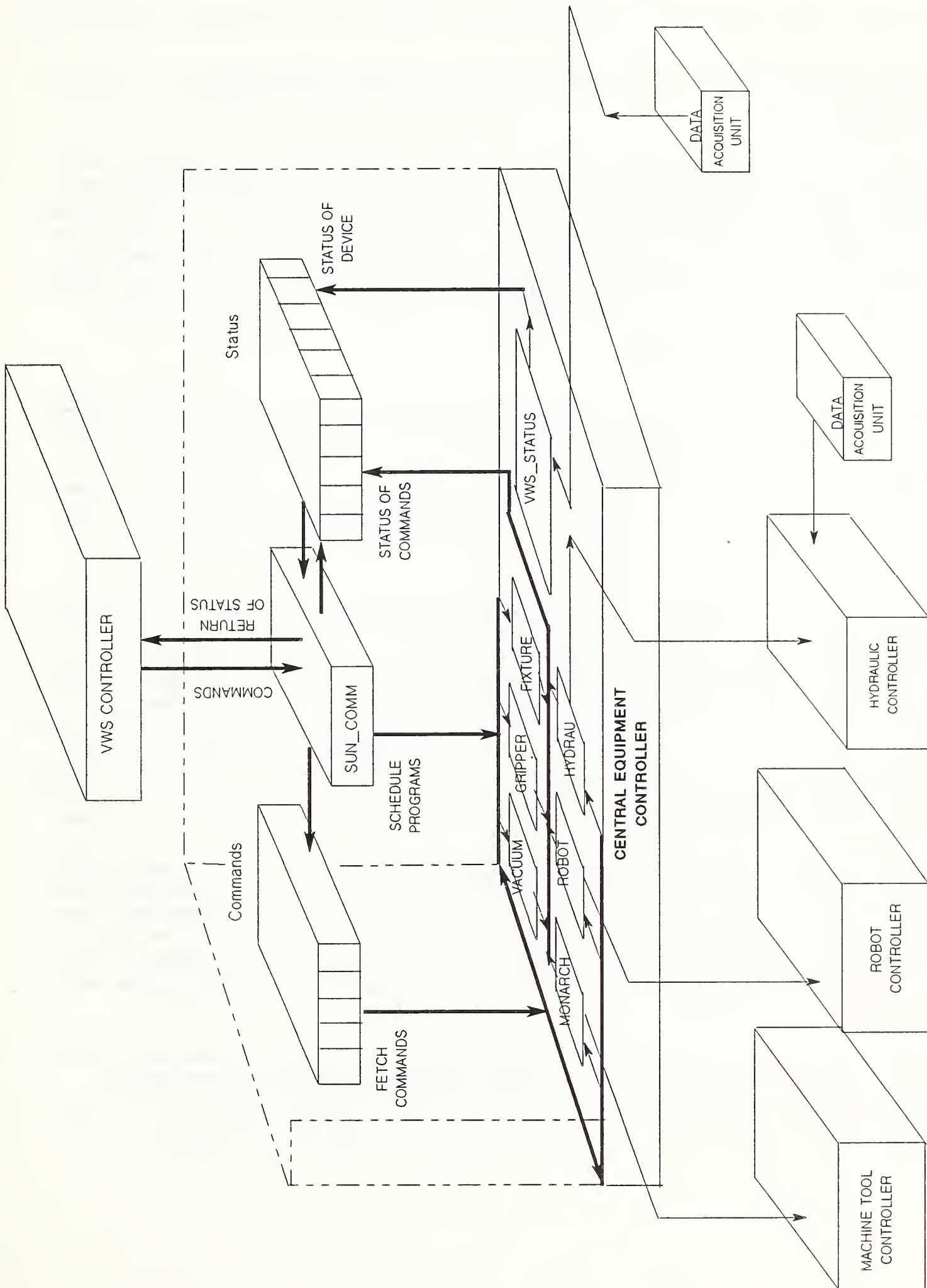


Figure 1 : CEC Programs and Other Controllers

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

acquisition units via an IEEE-488 interface, also called the HPIB. The other data acquisition unit is linked to the hydraulic controller via an IEEE-488 interface.

The VAL II robot controller is command driven, that is, robot programs do not reside in the robot memory but only in the WC. The WC packages each robot command into a command string which is transmitted to the CEC. After parsing the command string, the CEC activates one of its programs, which sends the robot command to the VAL II controller. Figure 1 shows how commands and statuses are routed from the WC through the CEC and dispatched to the various equipment controllers. A more complete description of the features and operations of the workstation controller is given in references. [KRA1] [KRA2] [KRA3] [K&J1] [NA&J].

The CEC can assess the status of seven work volumes within the workstation. Each of these work volumes is instrumented with sensors. Some of the sensors are simple micro-switches which are opened or closed by the motion of the delivery trays, positioning of the machine table, positioning of the workpiece, movement of the vise and the movement of the swing clamps. These micro-switches are connected to the digital voltmeter boards in data acquisition units and the statuses of these switches are determined by sensing the presence or absence of voltages across them.

The CEC makes an inquiry to the data acquisition units, whenever it needs to assess the status of a particular device of the VWS and sends an instruction to the data acquisition unit whenever it needs to activate a particular device. For example, the CEC can command the robot gripper to open, close or rotate simply by instructing the data acquisition unit to close switches which turn on pneumatic valves and servo motors that drive the gripper to the commanded position. In either case, the CEC will command a robot gripper operation only if it receives such a command from the WC. Similarly, the CEC will report the status of a physical device to the WC only when it receives a status request from the WC.

Figure 1 shows the relationship between the CEC and the two data acquisition units. One of the data acquisition units operates under the control of the CEC, while the other unit operates under the control of the hydraulic controller. All three computer-based controllers operate under the control of the CEC. It should be noted that the robot arm movements are under the control of the robot controller whereas the robot gripper motions are under the control of the CEC.

In the preceding chapter we discussed the implementation of four controllers on the floor of the VWS, with particular emphasis on the communication between them. The following sections treat the specific operation of each controller.

2.0 CENTRAL EQUIPMENT CONTROLLER (CEC)

The CEC is the HP-9000 series 520 minicomputer, with three central processing units (CPU) and 2.25 M-bytes of memory. The communication ports consist of three RS-232 interfaces and an HPIB Interface. These interfaces serve as communication links with other controllers. The operating system provides utility programs, including I/O routines, peripheral handling, and either the extended BASIC programming language or the HP-UX control language [HP1]. We are currently using the extended BASIC language which allows one to take full advantage of the multi-processor environment.

2.1 COMPUTER SYSTEM ORGANIZATION

The three CPU's provide for efficient program handling. This is especially important in the multi-tasking environment of the VWS. The memory of the CEC is divided into partitions. Each program runs independently in its own partition and interacts with other programs that are running in their own partitions. The computer resources are shared by these partitions. The multiple partitions allow the running of several programs simultaneously and allow more efficient use of multiple CPU's.

At the heart of the Model 520 is the Memory/Processor Bus (MPB). The MPB resides within the Memory Processor Module (MPM) assembly of the model 520. The MPM has card slots that accommodate up to 12 circuit cards, called finstrates. There are three types of finstrates: central processor, memory, and input/output processor. This model 520 is configured with three CPU's, 2 memory units and one input/output processor [HP1].

2.2 SOFTWARE STRUCTURE

The CEC's software is a collection of eight different programs. Each program performs a specific task and is assigned to its own partition. This structure permits one to more easily make programmatic changes to the functional operation of each task and to include new tasks. Under this program structure, the CEC's software is divided into the following eight program functions [MAG]:

1. Communication with the workstation controller and the parsing and the dispatching of commands. The SUN_COMM program performs these functions.
2. Sensory interaction. The VWS_STATUS program periodically reads the sensors and operates the actuators.
3. Establishing communication with the robot controller and sending robot commands. The ROBOT program performs these functions.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

4. Establishing communication with the machine tool controller and loading, selecting and executing NC programs. The MONARCH program performs these functions.
5. Operating the gripper by opening, closing and indexing the rotary pads. The GRIPPER program performs these functions.
6. Operating the vise. (opening, closing) and reading the clamping force of the vise. The FIXTURE Program performs these functions. If the code for the command set is F, the FIXTURE program will be activated.
7. Operating the vacuum chip removal system. The VACUUM program performs this function.
8. Operating the hydraulically actuated fixturing devices. The HYDRAULIC program performs this function

A control program called VWS_CONTROL, creates the separate partitions, loads each program into its own partition and executes each program. For example the GRIPPER program that controls the operation of the gripper is activated by the following statement [HP 1]:

```
CREATE PARTITION GRIPPER; PRIORITY 3, TYPE "BASIC", EXECUTE "LOAD";  
"GRIPPER:CS80,7,0",1"
```

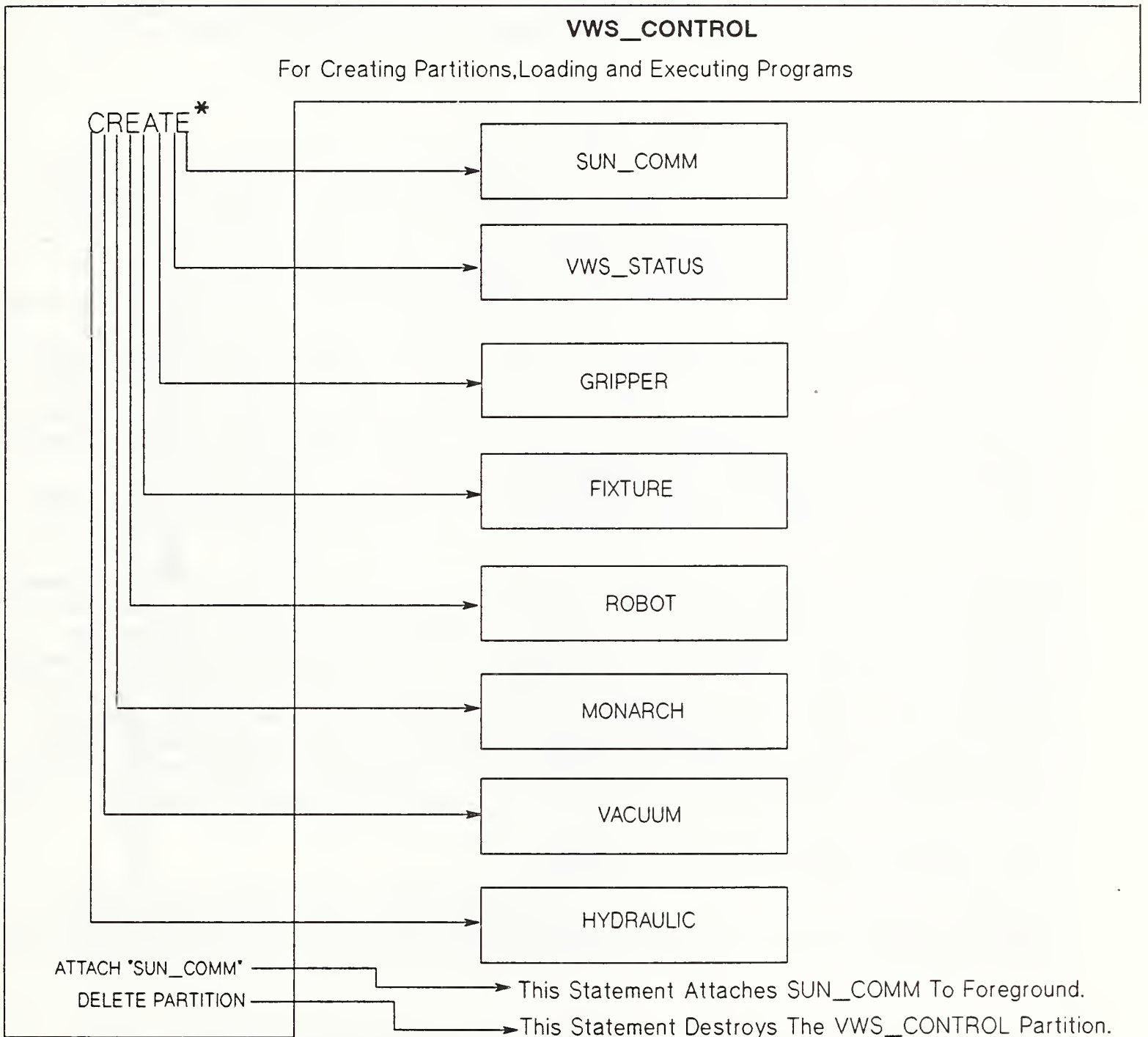
Other programs are loaded into their respective partitions by executing similar statements within The VWS_CONTROL program. Figure 2 illustrates how The VWS_CONTROL program creates eight different partitions, loads a program into each partition and executes each program. After VWS_CONTROL finishes these tasks, it deletes its own partition. The SUN_COMM and the VWS_STATUS programs continue to operate in the active run state, while the last six programs shown in Figure 2, are suspended in the WAIT state.

After each program is loaded, it can be executed either in the foreground or the background. However, only one program can run in the foreground at one time. In order for a program to have access to the CRT and the keyboard, its partition must be made a foreground partition. The VWS_CONTROL uses the statement, ATTACH SUN_COMM, to run the SUN_COMM program in the foreground and thus give it access to the CRT and the keyboard. All other partitions are background partitions. However, any background partition can become a foreground partition, if some other partition executes the ATTACH statement.

2.3 SOFTWARE OPERATING FEATURES

The SUN_COMM program serves as an executive program to seven other programs that are operating in their separate partitions. All of them except the VWS_STATUS program operate in a wait state until each program receives a command and a synchronization signal from the PARSER subroutine of the

FIGURE 2. OPERATION OF VWS_CONTROL PROGRAM



* An Example of Syntax:

CREATE PARTITION "SUN_COMM"; PRIORITY 3, TYPE "BASIC", EXECUTE "LOAD" "SUN_COMM:CS80,7,0", 1

SUN_COMM program. Figure 3 shows the relative scope of influence of the subroutines within the SUN_COMM program. The upper left column of this figure shows some commands which are sent from the WC to the CEC to obtain the status of the commands that were transmitted to the VMC, the robot and other devices. A command status informs the WC whether the CEC has executed a particular command. To confirm that a device performed as it was commanded, the WC sends one of the commands listed in the lower left of Figure 3.

2.3.1 Communication With Workstation

The WC initiates all communication with the CEC. The CEC only sends status messages to the workstation computer when it receives a request from the workstation. The only exception to that rule is to let the WC know about the outcome of the communication integrity check after receiving each transmission. Both the CEC and the WC perform integrity checks on the messages they receive. To do this, a block check character (BCC) is attached to each message string, and an immediate acknowledgement is sent to the sender, if the calculated BCC on both ends of the communication link agree. At the CEC end, the BCC is converted to a three digit decimal number prior to transmitting it to the WC. If the BCC transmitted equals the BCC received, the integrity of the communication link is acknowledged with an "A"; if not, a negative acknowledgement is transmitted with an "E". A negative acknowledgement causes a re-transmission of the command string [MAG]. The communication integrity task is performed by subroutines BCC and Hp_to_sun.

The workstation controller always requests the status of the previous command before issuing a new command. The central equipment controller responds immediately to the workstation's status request by returning either the status of a physical device or the status of the individual commands from the command sets. The three choices for reporting the command status are: (1) the command is still executing, (2) the command was successfully executed, and (3) the command was unsuccessfully executed. In the latter case, the reason for the unsuccessful execution of the command is transmitted. If the command was successfully executed, a new command is issued by the WC.

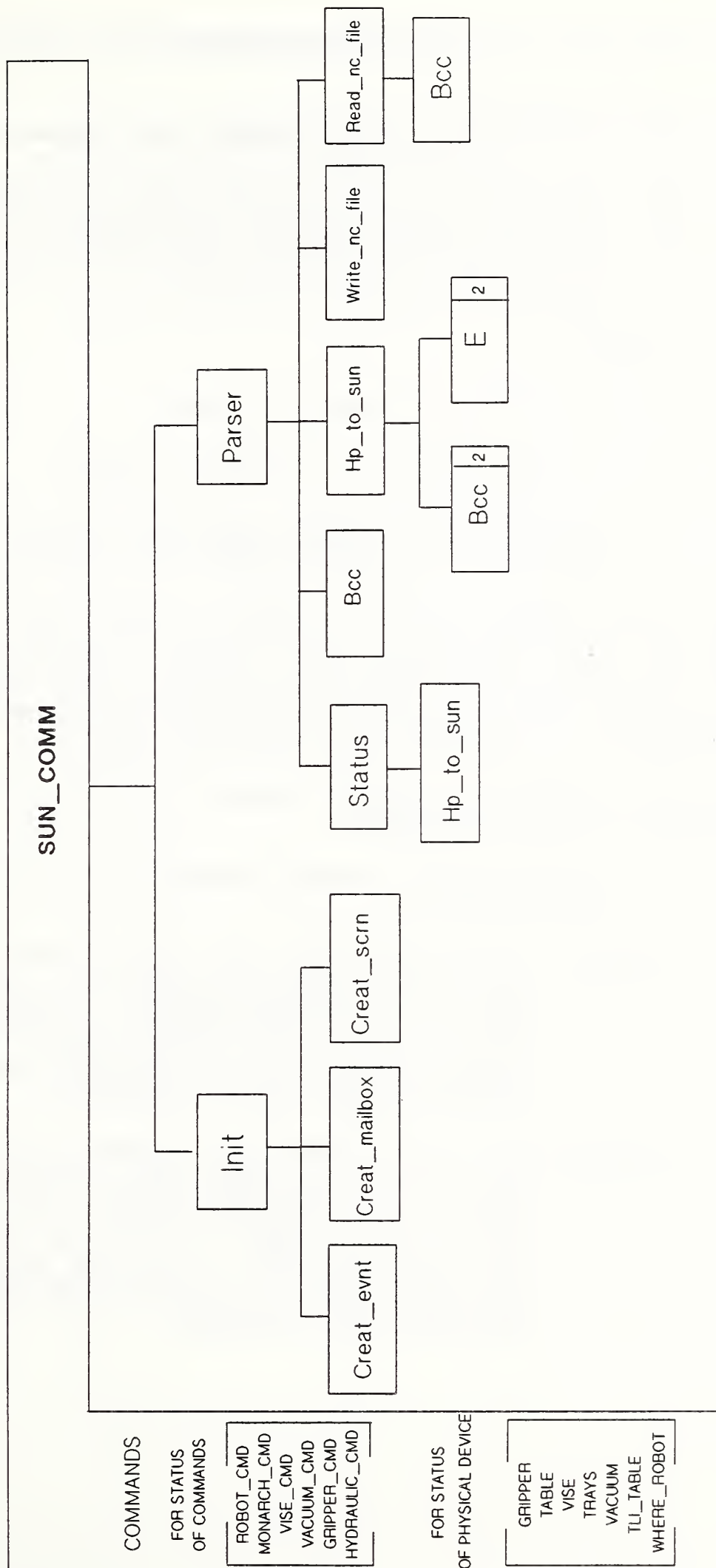
2.3.2 Structure of the Command String

Several commands may be sent in a string from the workstation controller. These are received at the central equipment controller in the following form:

*A*M1*M2.....Mn*B

This command string is decoded by the subroutine Parser at the central equipment controller to extract the following meaning [MAG]:

FIGURE 3. DIAGRAM SHOWING RELATIONSHIP BETWEEN SUN_COMM AND ITS SUBROUTINES



THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

- A - a four digit sequence number with leading zeroes,
- Mn - the individual command sets within the command string and,
- B - the block check character (BCC) (0-255).

The individual commands, Mn, have the following format:

Mn = Cmmmmmb

- mmmm - a four-digit, command set number,
- b - a string of letters forming the command,
- C - a command set identifier, indicating one of the following command sets : R=Robot, M=Monarch, F=Vise, H=hydraulic, G=Gripper, V=Vacuum, S=Status, A=acknowledgement,

Two rules for the command string that must be followed are: (1) the command set identifier cannot be repeated in the command string and (2) when C=S, only status commands are contained in the command string. There is an exception to the latter rule when an NC program is downloaded to the CEC for eventual transfer to the machine tool controller [MAG1].

When the CEC is requested to provide status or transmit a file, it will reply in one of the following forms:

- (1) E The BCC for the message transmitted does not agree with the received BCC.
- (2) 05AjjjjB The BCC does agree. The leading "05" shows that there are five bytes in the string, not counting the BCC byte; A is an acknowledgement of correctly received string; jjjj is a four digit number, including leading zeroes, that gives the identifying number of the overall command string, and B is the BCC.
- (3) NNSmmmmLB This is the form of the response to the status command: where NN is a two digit number indicating the number of bytes in the string that follows, not including the Block check character. S identifies the message as a status message, mmmm is the local sequence number of the status command, L is the status message and B is the block check character.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

- (4) LB L is a line of NC machine code being loaded from WC into CEC and B is the block check character.

Five examples of actual command sets are given in Table 1. These include examples of the Workstation Controller's command string and the corresponding replies from the CEC. An explanation of each command is also given in this table.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Table 1

Typical Workstation Operation: [MAG 1]
Remove a Part from the Vise and Place it in Tray #1

<u>Step</u>	<u>Workstation Controller Command String</u>	<u>Equipment Controller Reply</u>	<u>Explanation</u>
1.	*0054*R0009D0 Move MV1: TRANS(63.5,0,0,90,-90,0)* G0004OPEN*05A0054		Request the robot to move to a point 300 mm above the center of location 3 of tray #1. Simultaneously open the gripper's fingers.
2.	*0055*S0039ROBOT_CMD*	S00390009DONE	Indicates that the robot has successfully completed robot command 9. If the command number were 8 this status command would be repeated until the message regarding robot command 9 appeared. If the message associated with command 9 were different from "DONE", the workstation controller would enter an error reporting mode and the implementation of the following commands would not take place until the error condition was resolved. This logic is implicit for all subsequent status responses.
3.	*0056*S0040GRIPPER_CMD*	S00400004DONE	Gripper command 4 executed successfully.
4.	*0057*R0010DISABLE_TRAY1*	05A0057	Set software interlock flag to disable MHS from accessing tray #1 until it is released by a RELEASE-TRAY 1.
5.	*0058*S0041ROBOT_CMD*	S00410010DONE	Tray #1 movement locked out.

*NOTE: Only a few of the operations required to remove a part from the vise and place it in tray #1 are given in this table. There are additional operations given in Table 1 of "Vertical Machining Workstation of AMRF: Equipment Integration" [MAG 1].

2.3.3 Parsing Commands

One of the major functions of the SUN_COMM program is to decode the command string received from the WC. This task is performed by a subroutine called Parser. The operation of PARSER is based on two assumptions: (1) Each command set identifier may be used only once in a command string. (2) the workstation controller must request and receive from the CEC the status of the previous command before another command containing the same command set identifier can be issued. This status is reported to the WC as either a "DONE" or an "ERROR".

The Parser subroutine receives the command string and decodes the individual commands according to rules given in section 2.3.2. Figure 4 shows a diagram which displays the control logic used by the parser subroutine to distribute commands to the appropriate mailboxes. The string variable B\$(7) in this Figure is equivalent to the command set identifier C of section 2.3.2. The individual commands are distributed to designated random access memory files. These memory files shown in Figure 4 are referred to as mailboxes. The dispensing of the commands to the appropriate mailbox will depend on the string variable B\$(7). For example, if B\$(7) takes on the value "R", the command is dispensed to one of the mailboxes designated for the robot. However if B\$(7) takes on the value of "G", the command is distributed to a mailbox designated for the Gripper. Then the value found in B\$(7), the first prefix to the individual command, is used as a key for distributing the commands to the appropriate mailbox.

2.3.4 Scheduling Event

Figure 4 shows how the SUN_COMM program serves as an executive program to six other programs. It activates one of the six programs from the wait state depending on the command it receives from the WC. The SUN_COMM program does this by executing one of the CAUSE EVENT statements shown in Figure 4.

The EVENTS shown in Figure 4 were created in subroutine Creat_evnt. These EVENTS are signaling devices used by cooperating programs in two or more partitions. An EVENT is a system variable which can be incremented, decremented or queried [HP 1]. We use EVENTS to synchronize processes in partitions and to schedule programs. Figure 3 shows that the Creat_evnt subroutine is one of four subroutines which are executed by the SUN_COMM program during the initialization process. This process begins when SUN_COMM calls subroutine Init.

Before an EVENT is scheduled, it must be created [HP 1]. All events, with the exception of Event "OK_TO_PROCEED", are created in subroutine Creat_evnt. This subroutine is a sequence of BASIC statements of the following type:

```
CREATE EVENT "ROBOT"
```

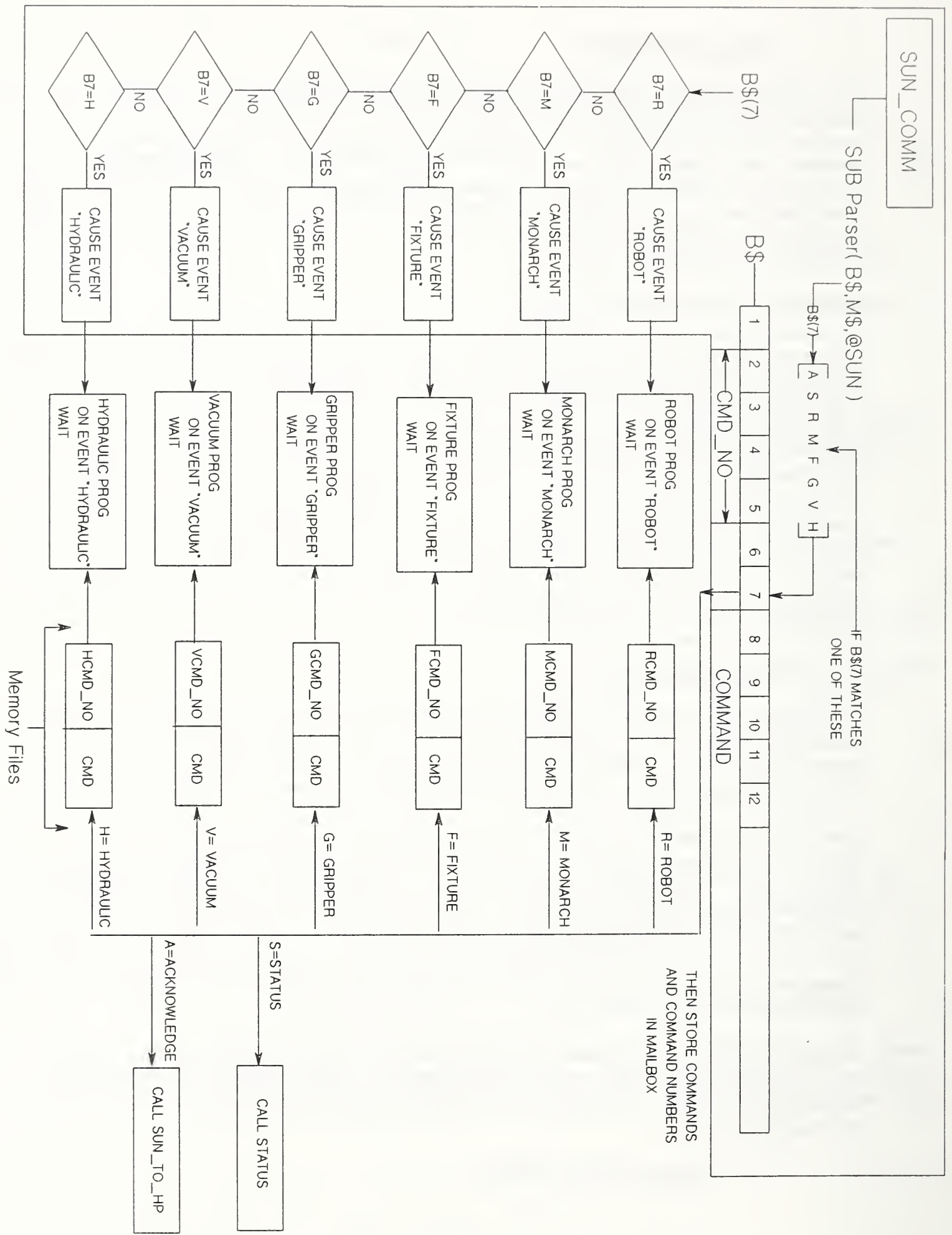


FIGURE 4. A SCHEME FOR ACTIVATING PROGRAMS

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Once an EVENT is created, any partition can schedule the EVENT by executing the CAUSE EVENT statement. Most EVENTS are scheduled by the Parser Subroutine of the SUN_COMM program.

The "OK_TO_PROCEED" EVENT is created in the "VWS_CONTROL" program. This event is handled differently because it is necessary to ensure that all required partitions and EVENTS are created before activating a particular EVENT. A pair of statements are required to activate an EVENT. First, a CAUSE statement in one partition schedules the event and second, an ON EVENT in another partition activates the event. Although these two statements are in different programs, they work together to achieve synchronization between programs. Unless the ON EVENT statement detects the execution of the CAUSE EVENT statement, the program remains suspended in a WAIT state [HP 1]. Upon the detection of the named CAUSE EVENT statement, the program continues to execute by jumping to a line number or performing a specified instruction, given in the ON EVENT statement.

2.3.5 Designating and Managing Mailboxes

The mailboxes exist as files residing in CEC memory. They are designated by using three BASIC statements: INITIALIZE MEMORY which prepares internal storage to act as mass storage; CREATE which enters the file name into the directory and specifies attributes of the file; and ASSIGN which gives an I/O path name to the file [HP 1]. A Command is written to the file by executing an OUTPUT statement. Prior to writing a particular file, the file is locked with a LOCK statement. This ensures against two partitions attempting to write to a file at almost the same instant, thus preventing one partition from inadvertently overwriting the data of other partitions. After writing to the file, the program executes the UNLOCK statement to release the file to other programs that are operating in other partitions. The use of the LOCK and UNLOCK features keeps each partition from ruining the work in-progress that is being performed by other partitions.

Subroutine "Creat_mailbox" is used to create all memory files. As an example, the memory file CMD_IN stores the most recent command and command number for each command set. This file is created with this BASIC statement, CREATE DATA"CMD_IN: MEMORY,0,0",13. Other memory files are created with similar statements.

2.3.6 Obtaining Statuses

The workstation controller requires two types of status reporting from the central equipment controller: (1) reporting on the status of commands and (2) reporting on the status of the equipment. Both of these are performed by a subroutine Status which is part of SUN_COMM, shown in Figure 5. When the Workstation Controller requests the status of a command, the SUN_COMM program calls the subroutine Status which in turn retrieves the requested status from one of the six mailbox locations, ranging from location 34 to 39. Six programs write their particular command-status to their designate mailbox

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

locations. For example, if the S\$ contains the command ROBOT_CMD, the ROBOT program would have already written the status of the last executed command to mailbox location 34. After reading the status of either a command or a physical device from a mailbox, the result is placed into a string variable Speed\$. The diagram in Figure 5 shows how the SUN_COMM program interacts with subroutine STATUS to retrieve the status of a command.

To get the status of a device, the Status subroutine reads the designated mailbox location for that device. These mailbox locations range from location 1 to 14. The "VWS_STATUS" program which runs in its partition will periodically write the status of the equipment to 14 mailbox locations. This program updates these memory locations every 20 ms. Figure 5 shows how the SUN_COMM program interacts with the VWS_STATUS program to retrieve the status of a physical device. Figure 6 shows the relationship between the VWS_STATUS and its subroutines. The VWS_STATUS program retrieves the voltage and resistance readings from the SENSORS subroutine which in turn calls the Hp3497 subroutine. The Hp3497 subroutine causes the data acquisition unit to read the voltage and ohmic levels from all of the sensors. The values of these readings are placed into a sixteen element array B. These elements of the array B contain: (1) the ohmic readings for the vise's force sensor and (2) the voltage readings for the potentiometers on the vise's gripper. The clamping force of the vise is obtained by multiplying ohmic readings by conversion constants.

In addition to reporting the status of the equipment to the workstation controller, the VWS_STATUS program turns actuators (solenoid, pneumatic valve) on or off in response to commands from the workstation controller. To do this, Figure 6 shows that the VWS_STATUS program calls a subroutine ACTUATOR that calls another subroutine Hp_act to open/close all solenoids on the pneumatic line, according to the parameters found in the command file. These parameters identify a channel number and an operating condition to open or close the solenoid.

2.3.7 VWS Rules for Issuing Commands

Before the workstation controller issues a new command, it requests the central equipment controller to report the status of the previous command. For example, to request the status of robot command #9, the WC issues the command "*0055*S0039ROBOT_CMD" to the CEC. If the command is completed, the CEC will send the following status message "S00390009DONE". The WC interprets this status message to mean that the robot has completed robot command #9. However, if the last successfully executed robot command was #8, the workstation controller would re-issue the status command until the command number 9 appears in the reply message from the equipment controller [MAG1]. If the status word associated with reply message #9 from the CEC is other than DONE, the WC would enter an error reporting mode. The status word DONE means that everything in a software sense that pertains to the particular command has been successfully executed. After receiving the

FIGURE 5. Interaction of SUN_COMM and VWS_STATUS Programs

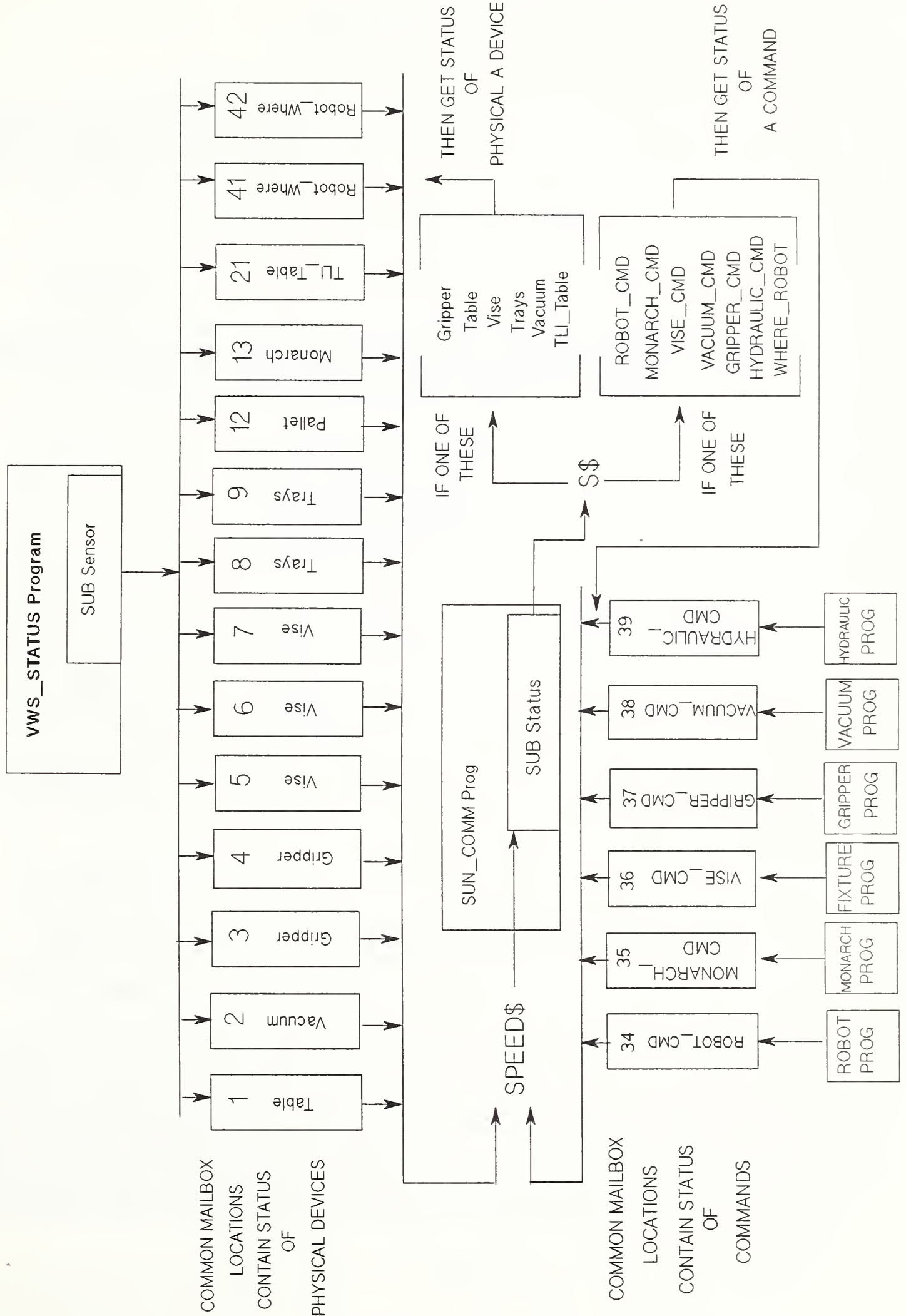
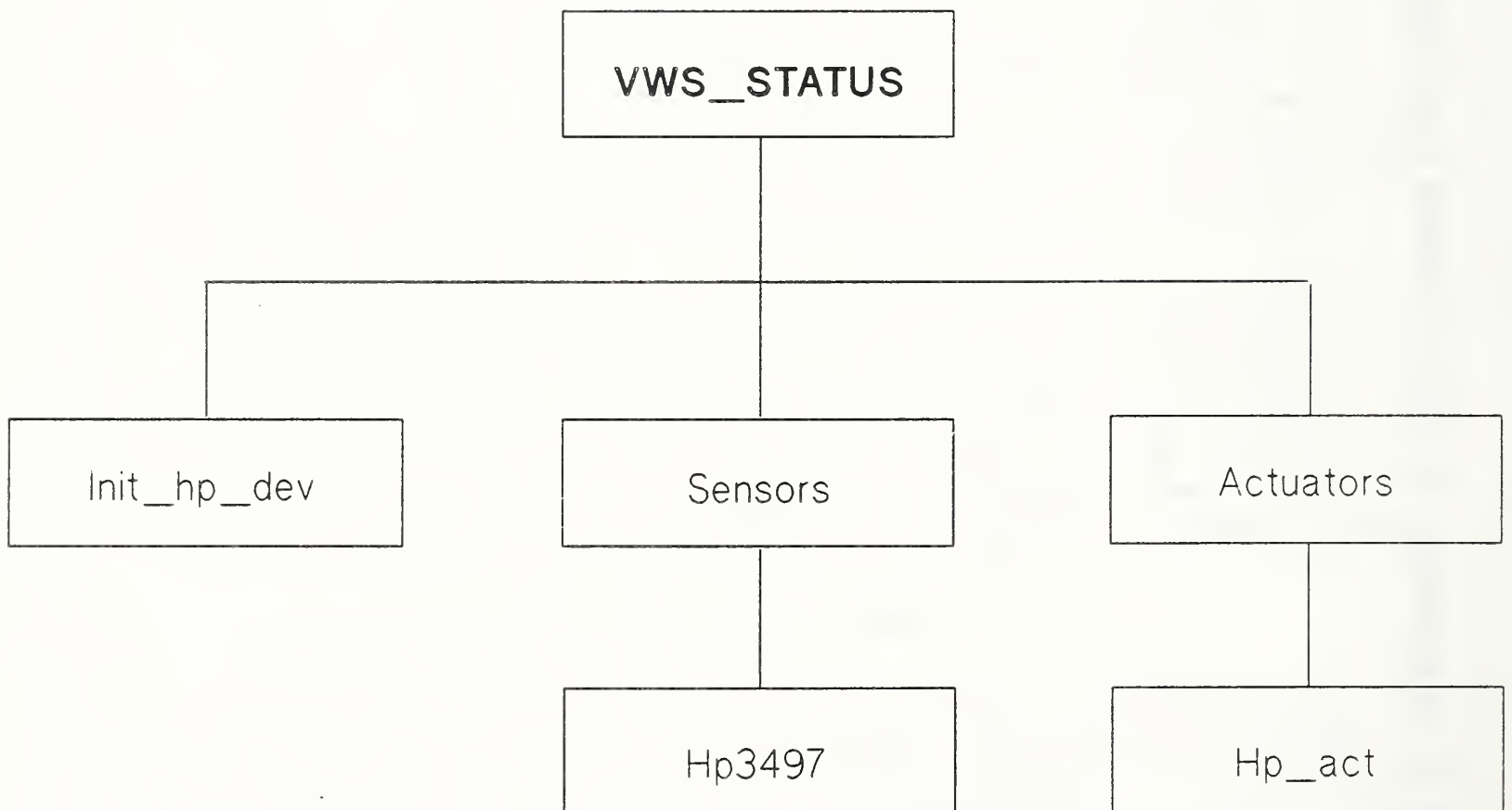


FIGURE 6. A DIAGRAM SHOWING RELATIONSHIP BETWEEN VWS_STATUS AND ITS SUBROUTINES



THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

status word DONE, the WC would then issue the appropriate status command to verify that the physical device actually performed as commanded. Some of the status commands that verify the status of the physical devices are listed in the table below:

Table 2: Equipment Status Commands
[MAG1]

<u>COMMANDS</u>	<u>REPLY</u>
GRIPPER	Gripper is opened and Indexed N Deg
TABLE	Table in park position, or Table not in park position
WISE	Vise jaws are d mm apart and Vise is locked or Vise is unlocked Part in vise, or part not in vise
TRAYS	Tray #1 in position and Tray #2 not in position Tray #1 not in pos. and Tray #2 is in pos.
VACUUM	Vacuum is on or Vacuum is off
MONARCH	Active or Idle
WHERE_ROBOT	$N_1/N_2/...../N_3$

NOTE: N_1, N_2, \dots, N_3 are the six joint coordinates giving the location of the robot arm's mounting flange for the gripper.

2.3.8 Software Safety Interlocks

Software safety interlock decisions are sprinkled throughout the various programs of the CEC. However, they are mainly found in the ROBOT and the MONARCH programs. The purpose of software interlocks is to prevent situations that will cause damage to the equipment or to the workpiece. There are numerous combinations of operations which may be potentially unsafe. Safety interlocks at the CEC level were implemented for the more obvious combinations of operations that may lead to dangerous situations. The remaining safety interlocks are implemented in the workstation

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

controller's work elements [ref Jay Jun]. This approach provides for a reasonably safe operation while still maintaining sufficient flexibility at the WC level.

The software interlock procedures were programmed to work in the following manner. Consider two mailbox locations M1 and M2 and two independent operations A1 and A2 that cannot be allowed to occur simultaneously. When operation A1 is to be performed, a message F1 is written to a mailbox M1. Similarly, when operation A2 is performed, a message F2 is written to a mailbox M2. Consider the procedure for activating an operation A1. After writing the message F1 to mailbox M1, the program reads mailbox M2 to determine the state of F2. If the message F2 indicates that the operation A2 is in progress, this operation A1 will not be allowed to start. The message F1 is withdrawn from mailbox M1 and after a short pause F1 is placed into mailbox M1 again. The program reads M2 again and re-evaluates message F2. This process is repeated about four times every second until either the message F2 indicates that operation A1 can start or the request for A1 has timed out. This same procedure takes place for operation A2. Consequently, operation A1 and A2 can never occur at the same time. [MAG1].

The above software interlock procedures are applied for the following operations:

- 1a. The robot cannot approach the machine tool table unless the machine tool is idle and the table is in parked position.
- 1b. A machine tool NC program cannot be started unless the robot has released the machine tool table.
- 2a. The robot cannot approach the parts tray unless the material handling system has released the tray.
- 2b. The material handling system cannot move a parts tray unless the robot has released that tray.
- 3a. The vise cannot open unless the machine tool is idle and the machine table is at the Parked position.
- 3b. A machine NC program cannot be started unless the vise has released the machine table.

An interlock command is used to give the robot or other types of equipment exclusive access to a work volume. No other equipment can have access to the work volume during the same time that the work volume is "locked" to the robot. For example, a safety interlock is used to give the robot exclusive access to tray #1 by disabling the Material Handling System's (MHS) access to the roller table that contains tray #1. Table 3 shows the commands to activate and to deactivate the interlocks of the MHS trays and the VWS table.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

TABLE 3: Software Interlock Commands
Ref: [MAG1]

<u>COMMANDS</u>	<u>EXPLANATION</u>
DISABLE_TRAYN	A software interlock command that disables MHS access to the tray by locking out any movement of tray N (N=1 or 2) until released with a RELEASE_TRAYN command.
RELEASE_TRAYN	A software interlock command that enables MHS access to the tray.
DISABLE_TABLE	A software interlock command that locks out any movement of the machine tool table until released with a RELEASE_TABLE command
RELEASE_TABLE	A software interlock command that cancels a previously issued, DISABLE_TABLE interlock command. This command returns control of the machine table to the machine tool controller

All four of these interlock commands are also included in the command set of the ROBOT program. The subroutine, Disable_tray, executes the two interlock commands that pertain to the two trays and the subroutine, Disable_table, executes the interlock commands that pertain to the machine table.

The last two commands are also part of the command set for the FIXTURE Program. The DISABLE_TABLE is issued to ensure that the machine table does not move from the park position while the vise is opening or closing. When the vise operation is completed, the RELEASE_TABLE command is issued to return control of table motion to the machine tool controller

A variation of the last interlock command RELEASE_TABLE is also executed in the MONARCH program. This interlock command permits other systems to have access to the machine table. Note that this meaning of the RELEASE_TABLE command is different from the one listed in the above Table 3. When the RELEASE_TABLE command is executed in the MONARCH program, the GE 2000 relinquishes control over the machine table and the machine table is locked at the park position. This sense of the RELEASE_TABLE command has not been implemented at the workstation controller level, but the other capabilities listed in Table 3 have been [MAG1].

3.0 GE 2000 CONTROLLER

The General Electric Mark Century (GE 2000) controller is used as the controller for the vertical machining center. In addition, this controller supports the inspection probe and the tool length indicator (TLI) options for the VMC. This probing capability is used in all NC programs which are

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

generated by the workstation controller. The NC program loads and executes a probing cycle routine, which ascertains the coordinate origin of a blank part, or locates the coordinates of a partly machined feature prior to performing additional machining.

The TLI feature is used to determine the length of each tool that will be used during the machining process. The CEC is capable of sending a request to the GE-2000 controller to perform a TLI function after each set of tool exchanges or after the machine's W axis has been moved. The TLI and the table-upload features are implemented to a limited degree at the workstation control level. The reader should refer to Kramer [KRA4] for more information.

The MONARCH program supervises all activities of the GE-2000 controller and this program interfaces the GE-2000 Controller to the CEC. The hierarchy of control for all subroutines within the MONARCH program is shown in Figure 7. The MONARCH program processes eight commands which originate from the workstation controller. These commands are listed in the subroutine Mon_cmd in Figure 7, and the corresponding subroutines which execute these commands are shown to the right.

The following sections describe: (1) the CEC to GE-2000 communication interface, (2) the communication protocol for CEC and the GE-2000 controller and (3) the MONARCH program that is used to process these VMC commands, which originate at the WC and are transmitted to the CEC. The subroutine that executes each command and the corresponding communication dialogue that occurs between the CEC and the GE-2000 controller are described. Each of the MONARCH commands has a corresponding GE-2000 command message-code. This command message-code is transmitted to the GE-2000 controller by the appropriate subroutine of Mon_cmd corresponding to the command received via the SUN_COMM program. The command message-code is contained on the second line of the communication dialogue (see sec. 3.2.2).

All commands are first transmitted from the WC to the SUN_COMM program of the CEC. After the SUN_COMM program parses the command string and determines that there is a Monarch command contained in the command string, it activates the MONARCH program and writes the command to memory file called Mailin. The Mon_cmd subroutine assigns an I/O path name CMD_IN to this file and reads the command from this file. As shown in Figure 7, the Mon_cmd subroutine then selects the appropriate subroutine to execute this command. Sections 3.2.2 through 3.2.7 describe Monarch commands and the communication dialogues that occur between the CEC and the GE-2000 controller.

3.1 DNC COMMUNICATION INTERFACE

The Mark Century Controller (GE-2000) provides a Distributed Numerical Control (DNC) communication port. This DNC communication feature lets the GE-2000 controller act as a terminal to the central equipment controller. The interface provides the capability for the transfer of part programs, data

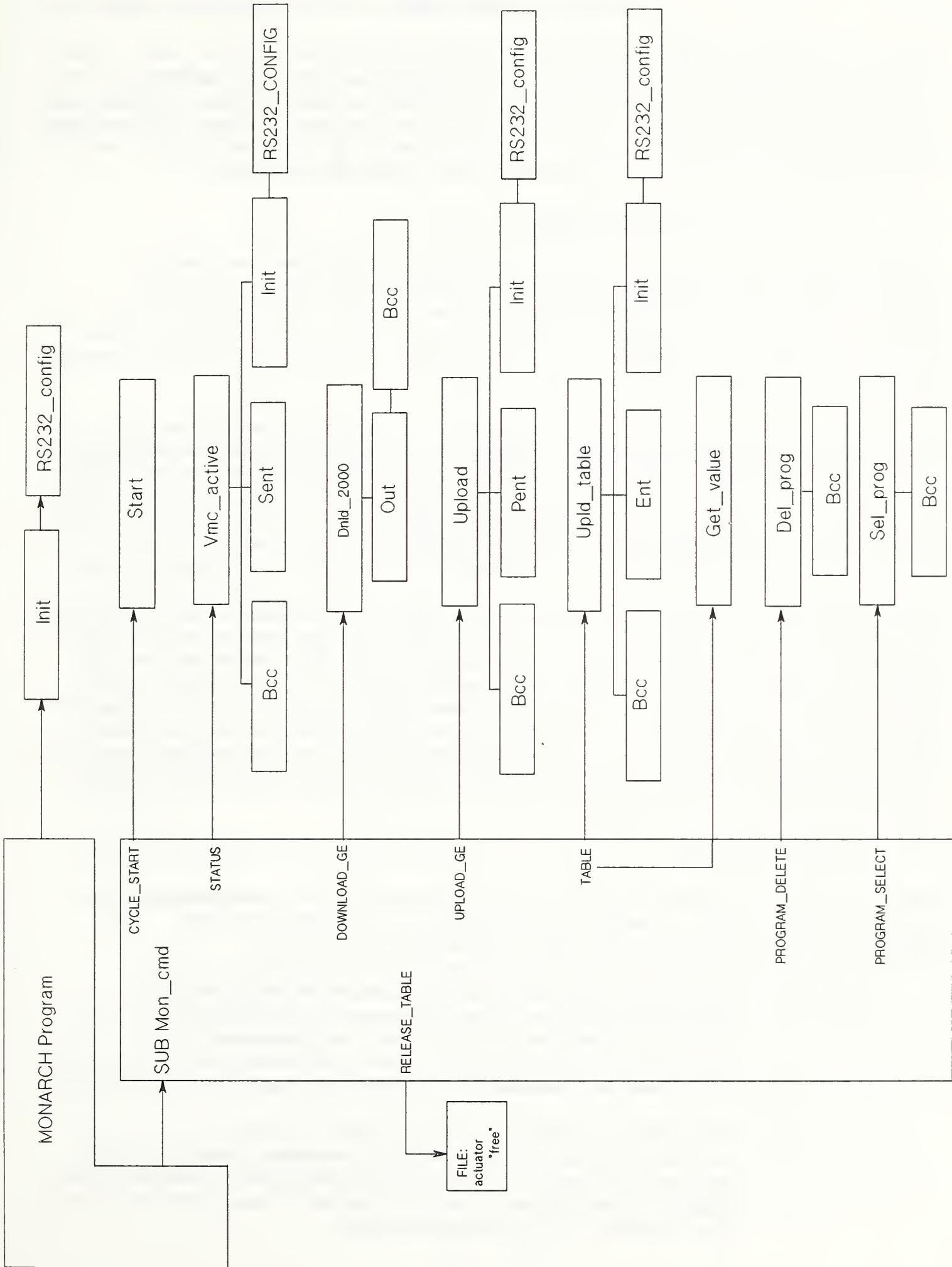


FIGURE 7. LEIGHTON DIAGRAM FOR SUBROUTINE "MON_CMD".

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

tables, and binary files between the GE-2000 controller and the CEC. Remote monitoring and status reporting are also supported by this interface in the GE-2000. The CEC communicates with the GE-2000 controller via a RS-232 link which operates at 4800 Baud. This RS-232 link physically plugs into the same RS-232 connector as that used for the tape reader on the the GE-2000, which ordinarily is the source of programs for NC machines [GEK1].

3.2 COMMUNICATION PROTOCOL

The DNC has two protocol levels, either one of which may be used for communicating with the host. The level 1 protocol provides no error detection or correction mechanism other than parity. This protocol is very simple to implement and may be the best solution for small shops without extensive programming support. This protocol is based on the RS-491 standard for tape reader/punch control [EIA RS 491].

The LEVEL 2 protocol is more sophisticated. It supports more functions, has better flow control to turn data streams on and off and provides for error detections and corrections. Its capabilities are similar to the ANSI X3.28 link protocol specifications [ANSI X 3.28-86]. Level 2 is the best solution for shops that have computer programming support to develop software. This Level 2 protocol is implemented in the "MONARCH" program which runs in its own partition on the equipment controller.

LEVEL 2 protocol is initiated when either end of the communication link sends an ENQ (dec 5) to initiate the bid to be master of the communication link. The first one that sends an ENQ becomes master of the communication link. However, if the GE-2000 initiates the communication, it will send an ENQ followed by a DEL character. The other end acknowledges its slave status by responding with a DEL and a "0". The master now transmits message blocks which, if correctly received, are acknowledged with alternate DLE 0 and a DLE 1 transmissions. After the final message block is sent and acknowledged, the master transmits an EOT [GEK1].

3.2.1 Command Message

Each transmission from the central equipment controller must begin with a Start-Of-Header (SOH, a decimal 1) followed by a command message. The command message consists of 3 ASCII characters as follows:

R/S The first character of the command which gives the Direction-of-Data. When the master transmits an R, the slave is commanded to take over as the master after the next End-Of-Transmission (EOT). When the master transmits an S, the slave is commanded to accept the text from the master.

N/C The second character of the command is either N or C. The N is sent to start Longitudinal Redundancy Checking (LRC). And the C is sent to start Cycle Redundancy Checking (CRC) [GEK1]. The LRC and The CRC provide a means for checking the efficacy of the communication.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

The third character of the command is decoded by the slave to set up for the desired operation. The possible options for this third character are as follows:

Command Character	Operation
S	Part Program Transfer
O	Table Transfer
Q	Binary File Transfer*
K	Remote Machine Control
L	Fixed Format Status Report
F	Upload GE 2000 Program Directory*
D	Delete File from GE 2000 Directory
E	Download Part Program and Execute*
*	Not used in our program

The commands in the MONARCH program, the corresponding Monarch subroutines and the resulting command message-code to execute the command are shown below.

<u>Monarch Command</u>	<u>Monarch Subroutine</u>	<u>GE-2000 Command Message</u>
CYCLE_START	Start	*
TABLE	Upld_table	RNO
STATUS	Vmc_active	RNL
DOWNLOAD_GE	Dnld_2000	SNS
UPLOAD_GE	Upload	RNS
PROGRAM_DELETE	Del_prog	SND
PROGRAM_SELECT	Sel_prog	SNK

*NOTE: There is no corresponding GE-2000 command message-code for this MONARCH command. the cycle start switch on the GE-2000 controller is hot wired and is directly controlled by the CEC start subroutine.

The general form and sequence of the message transmission is as follows:

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

<u>Master</u>	<u>Slave</u>
ENQ	
SOH COM-MSG STX ETX BCC	DLE 0
STX Text-1 ETX BCC	DLE 1
STX TEXT-2 ETX BCC	DLE 0
EOT	DLE 1

NOTE: Start-of-Text is denoted by STX
End-of-Text is denoted by ETX.

These sequences are described in detail for each message in sections 3.2.2-3.2.7

3.2.2 Part Program Download in Level 2 Protocol.

For a part program download, the command message is SNS or SCS. The Central Equipment Controller first sends an ENQ which switches the GE 2000 controller from the dialogue mode to the level 2 protocol mode. The overall sequence for this transmission is as follows:

Central Equipment Controller	GE 2000 Controller
ENQ	
SOH SNS STX ETX BCC	DLE 0
STX (part program data) ETX BCC	DLE 1
STX (part program data) ETX BCC	DLE 0
EOT	DLE 1

The task of downloading NC programs from the CEC to The GE-2000 controller is performed by the subroutine Dnld_2000, as shown above.

3.2.3 Part Program Upload in Level 2 Protocol

The NC part programs may also be transferred from the GE-2000 to the CEC. This task is known as uploading and is performed by subroutine Upload, in the MONARCH Program. After an NC program, has been uploaded, it is stored in a CEC memory file "NC_UP:MEMORY,0,3".

The command message for the part program upload is RNS. The associated dialogue between the CEC and The GE-2000 controller is as follows:

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Central Equipment Controller	GE-2000 Controller
ENQ	
SOH RNS STX ETX BCC	DLE 0
EOT	DLE 1
DLE 0	ENQ DEL
DLE 1	STX (prog data) ETX BCC
DLE 0	STX (prog data) ETX BCC
SOH SOH SOH	EOT DEL

If a file name is not included, a program that has been selected from the GE-2000 control panel is uploaded. If the file name is included, it must end with an extension of .PRO, .GSU, or MSU to specify type PROG, GSUB, or MSUB. [GEK1]

3.2.4 Select Program for Execution

The subroutine Sel_prog will select a GE-2000, resident program for execution whenever the command "PROGRAM_SELECT" is read from a file "CMD_IN:MEMORY,0,0" residing on the CEC. The file name for the program is read from same memory file, starting at position 16. The CEC transmits an ENQ to initiate its bid to become master of the communication link. After the CEC transmits an SOH character, it sends an SNK command message-code to the GE-2000 controller. This code causes the GE-2000 to recognize the file name, given on the command message line. The communication dialogue between the CEC and the GE-2000 controller is as follows:

<u>Central Equipment Controller</u>	<u>GE-2000 Controller</u>
ENQ	
SOH SNK STX file_name ETX BCC	DLE 0
EOT SOH SOH SOH	DLE 1

3.2.5 Delete Program from GE-2000 Memory

The subroutine Del_prog deletes a GE-2000 resident file, whenever the command PROGRAM_DELETE is read from the memory file "CMD_IN:MEMORY,0,0" of the CEC. Subroutine Mon_cmd, in the CEC, retrieves the command from the memory file and then calls subroutine Del_prog to carry out the command. This subroutine sends an SOH character followed by a command message-code "SND" to the GE-

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

2000 controller. This message code instructs the GE-2000 to delete the file designated by the file number on the command line. The "PROGRAM_DELETE" command deletes the program with the program-number "NNNNNN.PRO" or "NNNNNN.GSU". The dialogue between the CEC and the GE-2000 controller is as follows:

<u>Equipment Controller</u>	<u>GE-2000 Controller</u>
ENQ	
SOH SND STX file_no ETX BCC	DLE 0
EOT	DLE 1

3.2.6 Retrieve Tool Length Offset Table

Subroutine Mon_cmd retrieves the TABLE command from the memory file "CMD_IN:MEMORY,0,0". This subroutine then calls subroutine Upld_table to execute the Tool-Length-Offset command. The execution of this command transfers the content of the tool length offset table to the central equipment controller. The dialogue between the CEC and the GE-2000 is as follows:

<u>Equipment Controller</u>	<u>GE-2000 Controller</u>
ENQ	
SOH RNO STX "TOFFSET" EXT BCC	DLE 0
EOT	DLE 1
	ENQ DEL

NOTE : TOFFSET is a mnemonic for tool offset

3.2.7 Machine Tool Status

When the workstation controller sends a machine-tool status request to the CEC, the SUN_COMM program activates the MONARCH program, which in turn calls the Mon_cmd subroutine. This subroutine reads the STATUS request from the "CMD_IN" memory file and then subroutine calls the Vmc_active subroutine. This subroutine sends an ENQ to initiate its bid to become master of the communication link. After receiving DLE 0, the acknowledgement, the Vmc_active subroutine sends a SOH character followed by the "RNL" command message-code. This command message causes the GE-2000 to send the Active/Idle status to the CEC. The communication dialogue between the CEC and the GE-2000 is as follows:

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Central Equipment Controller

GE-2000 Controller

ENQ	
	DLE 0
SOH RNL STX ETX BCC	
	DLE 1
EOT	
	ENQ DEL
DLE 0	
	STX(status data)ETX BCC
DLE 1	

4.0 THE UNIMATION VAL II CONTROLLER

4.1 INTRODUCTION: THE CEC ROBOT PROGRAM

The VAL II controller that directs the robot arm is under the complete control of the CEC. When the workstation controller commands a robot task, it transmits a command string to the SUN_COMM program of the CEC. After the SUN_COMM program parses the command string and determines that there is a robot command contained in the command string, it activates the ROBOT program and writes the robot command to a memory file called Mailin. The Robot_cmd subroutine assigns an I/O path-name to this file and reads the robot command from this file. Figure 4 shows the scheme for activating the robot program and reading the robot command from the memory file and transmitting this command into the ROBOT program. After the Robot_cmd subroutine retrieves the command from the Mailin file, it passes this command to the Robot subroutine, discussed in section 4.4. The Robot subroutine assembles the command into a message package consisting of two headers, two sets of BCC characters and framing bytes. This message packaging is required to satisfy the multi-layer protocol required by the supervisory system of the VAL II controller. These protocols are discussed in sections 4.3 thru 4.3.4. After assembling the message, the Robot subroutine calls the Network subroutine that transmits the message over the RS-232 line to the supervisory port of the VAL II controller. The VAL II supervisory system parses the incoming message and identifies the message type.

After identifying the message, the supervisory system routes the message to a logical unit that handles that type of message. Message identification and logical units are described in section 4.3. The VAL II supervisory system and the CEC engage in an ongoing control-message dialogue during the transmission of a data message to or from the VAL II controller. Some of the message exchanges are required to control the communication link such as ACK, NAK, and REP. Other messages have more information such as DO MOVE SAFE, an instruction to VAL II or MOTION BEGIN and MOTION COMPLETE, two messages that carry status information from the VAL II controller.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Subroutines in the ROBOT program that handle these different types of messages are the Norm_val_msg subroutine and the Robot subroutine. These subroutines are described in section 4.4. All messages that are transmitted from the VAL II controller are received by the Normal_val_msg subroutine. If the message received by this subroutine is an informational or data type, the message is transferred to the Robot subroutine, which processes this information. However, if the message is a control type, the appropriate subroutine is called to transmit the required response to the VAL II controller. Sections 4.3.2. and 4.3.3 describe the different types of messages and section 4.4 describes the subroutines that handle different types of messages.

It should be noted that the operation of the gripper is separate and distinct from the operation of the robot arm. All commands for the robot arm are routed to the ROBOT program and then finally transmitted to the VAL II controller; whereas all gripper commands are routed to the GRIPPER program of the CEC. Subroutine Actuator in the VWS_STATUS program (Figure 5) reads these commands and subroutine Hp_act (Figure 6) finally sends a corresponding set of instructions to the data acquisition unit #1. This unit actually opens/closes a pneumatic solenoid on the robot gripper. For more information on VWS_STATUS program see section 2.3.6.

The VAL II Controller is a computer-based system that has a language specifically designed for programming and controlling the motions of industrial robots. All signals to and from the robot arm pass through the controller and are used by the controller to perform real-time calculations to control the arm movements and positions. The VAL II software is stored in a CMOS memory board, located in the controller. The software interprets the operating instructions for the arm. The VAL II controller transmits these instructions to the joint servo boards which control the robot arm movements. From the discrete encoders on the robot arm, the VAL II controller receives data about the robot arm position. In addition, the velocity feedback from the linear velocity transducers and the differential pressure feedback from the pressure transducers form a closed loop system within the VAL II controller [UNI1]. This information from the transducers never reaches the CEC.

The computer system for the VAL II controller is a DEC-11 system which contains a LSI-11/73 board, two DRV11-J interface boards, a Parallel I/O interface board, a CMOS memory board and two DRV-11-J quad serial boards [UNI1].

4.2 PROGRAMMING MODES FOR VAL II

The VAL II controller can be operated in three different programming modes: (1) robot control program, (2) process control program and (3) monitor command program [UNI2].

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

The VAL II controller is operated primarily in the monitor command mode. The central equipment controller communicates with the VAL II monitor by sending monitor commands via the RS-232 supervisory port. MONITOR is an administrative program in the VAL II controller that oversees the operations of the system. The VAL II monitor accepts monitor commands through the supervisory port and follows the instructions to direct the robot [UNI2]. It also initiates appropriate responses to monitor commands. The CEC dispatches instructions and commands to the VAL II monitor. Some robot motion instructions must be converted to monitor commands by preceding the instruction with the reserve word "DO". Normal monitor commands are not preceded by the reserved word "DO".

4.3 VAL II SUPERVISORY SYSTEM

The VAL II system communicates with the Central Equipment Controller (CEC) via its supervisory system, which includes a communication network within the VAL II controller. This communication network operates according to a rigorous communication protocol to ensure the integrity of the information being transferred between the VAL II system and the CEC. With the aid of this supervisory system the CEC can completely supervise the operation of the VAL II controller. In addition, the VAL II system can communicate with the CEC during program execution. The following functions are performed by the supervisory system:

1. Issues to the rest of the VAL II system any command from the CEC that can be typed at the robot terminal during normal operation. However, for our current robot driver routine, the command set is limited to those shown in Appendix I, part B.
2. Inputs and outputs data directly to and from programs running in the VAL II system. The users programs are those application programs that are not part of the operating system or the supervisor system of the VAL II system.
3. Monitors the operational status of the robot system.
4. Downloads and uploads user program and data. This feature is not currently being used.

When the supervisory communication link is connected and enabled [UNI2,RUD1], any combination of the first three types of communications can be dynamically selected and directed over the network. For example, when monitor commands are received over the network, the VAL II system provides additional information which allows the CEC to easily interpret the response to the command it transmitted. In particular, all responses are preceded by a number whose value differentiates normal responses from error messages. Each error message has a unique numeric code so that the error can be recognized easily. [UNI2]

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

The communication link between the central equipment controller and the VAL II Supervisory system can be separated into two categories: (1) the physical link which is the actual hardware that carries the messages and (2) the logical units which are the logical separation of the various communication elements of the VAL II system. Since VAL II uses only one physical link for its supervisory communication, it uses a coding scheme within every message to identify the type of communication. This scheme subdivides VAL II resources into separate subunits, referred to as "logical units" [UNI2]

The physical communication link for the supervisory communication is an EIA RS-423 serial line that is fully compatible with the EIA RS-232C Standard. The two are almost the same; the difference is the maximum allowable data rate with various lengths of cables. On our system, the communication data is transmitted at 9600 Baud, with a pattern of one stop bit for each 8-bit byte [UNI2].

The VAL II supervisory system is divided into five logical units. Each message contains an ID code that identifies the logical unit number (LUN) addressed by that message. The logical unit that receives a message can determine the nature of the message without discerning the physical path of that message. These LUN's are as follows [UNI2] .

<u>LUN</u>	<u>Logical Unit</u>
0	Network Manager
1	Short Status Information
2	Monitor Input and Output
3	Monitor Asynchronous Output
4	Program Terminal input and output
5	Disk input and output

Of these, our implementation of the supervisory interface software uses monitor input and output and monitor asynchronous output (designated as LUN 2, and 3.)

From the perspective of the VAL II system, LUN 2 and 3 work together to serve the role of the VAL II system terminal while the supervisory system has control of the VAL II system. This occurs when the NETWORK and SUPERVISORY system switches on the VAL II controller are enabled [UNI2,RUD1]. Under this condition, most VAL II monitor commands are directed to the Monitor Input and Output logical units.

4.3.1 VAL II Supervisory Protocol

Three layers of communication protocol are used to perform supervisory communication with VAL II logical units. These layers are referred to as the top-layer, the middle-layer and the bottom-layer. The top-layer of the supervisory communication consists of a number of tasks within the VAL II system. Each task performs transmissions and receptions of a particular type

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

of message. The possible types of transmissions are: (1) commands and messages from and to the monitor, (2) User program input and output (directed to the terminal), (3) disk commands and data transfers and (4) terse robot status information [UNI2]. Our implementation of this top-layer communication protocol includes tasks 1, 2, and 4.

All of the top-layer tasks perform their network communication activities by transmitting their output and receiving input through the middle-layer communication protocol. The middle-layer performs several functions:

1. It simultaneously accepts transmission requests from any top layer tasks and queues up messages for serial transmission over a single network line.
2. It accepts a serial stream of input messages from the network line and dispatches the input to the proper top-layer task.
3. It enforces a handshaking protocol that ensures that each request for action is properly dispatched before another request is accepted.

Functionally the middle layer protocol allows the transmission of different types of messages to be mixed on a single communication line. Even though a single communication line connects the VAL II system to the CEC; monitor commands, inputs, outputs and status information can be transmitted virtually simultaneously. For example even if the supervisory system is downloading the next instruction to be executed, it can still obtain status information regarding the current user program. [UNI2]

The bottom-layer of the supervisory communication protocol accepts a single transmission from the middle layer and simultaneously reads in a single message from the communication channel. Thus one side of the bottom layer communicates with the middle layer protocol and the other side deals directly with the hardware performing the communication. The most important task of the bottom layer is to ensure that each message transmitted is error free [UNI2]. To accomplish this, the bottom layer of the VAL II supervisory interface uses the Digital Data Communication Message Protocol (DDCMP) described below [DEC1].

4.3.2 DDCMP Protocol

DDCMP is a rigorous communication protocol that provides data integrity, message sequencing and management of the physical link. This protocol also defines the structure content and sequencing procedures for error detection and recovery. DDCMP is concerned with the logical transmission of data grouped into physical blocks known as data messages. The primary function of this protocol is to exchange the data messages while ensuring the correct sequencing and the integrity of the data messages when sent over the communication link [DEC1].

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

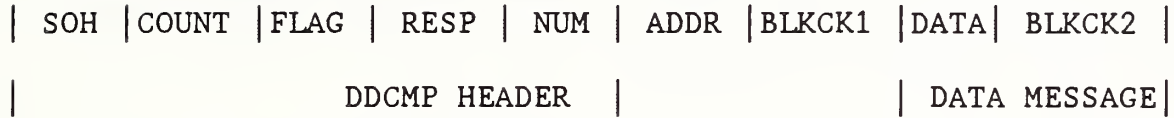
The DDCMP has a message exchange component that creates a sequential error-free link. This component transfers the data correctly and in the proper sequence over the physical link, which has some possibility of introducing errors. Once framing (padding characters for separating messages) is accomplished, this component operates at the message level, exchanging data and control messages [DEC1].

DDCMP is a positive acknowledgement protocol. For each message correctly received from the TOP-LAYER and passed to the CEC, a positive acknowledgement is returned on the link notifying the VAL II controller of the correct receipt of the data message. If incorrectly received, the data message is not passed to the CEC and is not acknowledged. Eventually, the message will be retransmitted. The DDCMP uses the Cyclic Redundance Checking (CRC-16) for error detection [DEC1].

There are five types of DDCMP control messages and one DDCMP data message. Figures 8 and Figure 9 show the formats for the DDCMP data message and the DDCMP control messages. The DDCMP data message has a field which contains the VAL II data message. Some other fields of the data message contain the message number, the BCC character, and the message number of the last correctly received message. These fields are shown in Figure 8. The unnumbered control messages shown in Figure 9 carry channel control transmission status and initialization notification between protocol layers. The five types of control messages, along with their explanations are given in Table 4 shown below.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

FIGURE 8 : FORMAT FOR NUMBERED DATA MESSAGE
Reference[DEC1]



SOH The identifier of numbered data messages

COUNT A fourteen bit number that specifies the number of 8-bit bytes in the data field. COUNT is combined with FLAG to form a 2-byte number

FLAG A two-bit number that is used to control link ownership and Synchronization. Flag is combined with COUNT to form a 2-byte number.

RESP A one-byte response number which is used to acknowledge correctly received messages. It is the number of the last correctly received message.

NUM A one-byte transmit number which is used to denote the number of the data message.

ADDR A one-byte number used to denote the address of a station on a multi-point link. For a point-to-point link the address value is 1

BLKCK1 A two-byte block check on the message header

DATA The message data field containing COUNT bytes of data

BLKCK2 A two-byte block check on the data field

NOTE: The data portion of the message flows into and out of the VAL II .

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

FIGURE 9: GENERAL FORMAT FOR UNNUMBERED CONTROL MESSAGE
Reference [DEC1]

| ENQ | TYPE | SUBTYPE | FLAGS | RCVR | SNDR | ADDR | BLKCK3 |

- ENQ A one-byte number that identifies the message as an unnumbered control message. It always has a value of decimal 5.
- TYPE The control message type, this value denotes each control message.
- SUBTYPE A 6-Bit quantity. It provides additional information for some message types. Its use is specific for each message type.
- FLAGS The FLAGS are two-bit quantities. They are the same as described in Figure 8.
- RCVR This is the control message receiver field. It is used to pass information from the data message receiver or slave station to the data message sender or master station. Its use is specific for each control message type.
- SNDR This is the control message sender field. It is used to pass information from data sender or master to the data message receiver or slave. Its use is specific for each control message type.
- ADDR ADDR is the station address field. It is the same as described for the numbered data message field shown in Figure 8 .
- BLKCK3 This is the block check on the control message. BLKCK3 is computed on fields ENQ through ADDR, using cycle redundance check (CRC-16) techniques.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Table 4: TYPES OF DDCMP CONTROL MESSAGES
[DEC1]

<u>Control Message</u>	<u>Explanation</u>
Acknowledge (ACK)	The ACK message is used to acknowledge the correct receipt of a numbered data message
Negative acknowledge (NAK)	The NAK message is used to pass error information from the data receiver to the data sender. The reason for the error is included in the subtype fields of the control message.
Start (STRT)	The STRT message is used to establish initial contact and synchronization of the link. It is used only on startup or re-initialization. It operates in connection with the start acknowledge message STACK.
Start Acknowledge (STACK)	The STACK message is returned in response to a STRT when a station has completed re-initialization and has reset the message number.
Reply to message number(REP)	The REP message is used to request the received message status from the data receiver. It is usually sent when the receiver has transmitted a data message and has not received a reply within a timeout period.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

The sequencing of message exchanges between the VAL II controller and the CEC for three different operating modes, namely start-up mode, data exchange mode, and re-initialization mode is shown in Table 5. The start_up mode is activated immediately after the VAL II software switches NETWORK and SUPERVISORY are enabled. These switches are enabled by typing the command "ENABLE NETWORK,SUPERVISORY", at the VAL terminal. The VAL II supervisory system remains in the start-up mode and continues to transmit STRT until it receives a STACK from the CEC. When this happens, the VAL II controller switches to the data-exchange dialogue mode, shown in Table 5. The data message transmitted from the VAL II contains the LUN in the first field of its VAL HEADER, shown in Figures 10 and 11. After the CEC receives this data message, it responds to the VAL II by sending an ACK and by incrementing the response number in the DDCMP HEADER. This action acknowledges the correct receipt of the data message. After acknowledging the VAL II data message, the CEC can either transmit a command or a reply to the VAL II controller. A reply is transmitted by the CEC only if the data message from the VAL II controller originated from LUN 3 [UNI2]. Otherwise, the normal response is to either transmit a command or an ACK. To send a reply, the LU2 routine within the Robot Subroutine calls the Reply Subroutine. Figure 12 shows that this Reply Subroutine transmits the reply message to the VAL II controller, via RS-232 link.

Re-initialization is a transitional communication process that occurs between consecutive commands. After the execution of a command is completed, The CEC sends a STRT to the VAL II controller. The communication dialogue of Table 5 shows that the VAL II responds by sending a STACK followed by a series of STRTs. It continues to send STRTs until the ROBOT program receives another robot command from the SUN_COMM program. At this time, the LU2 routine of the Robot subroutine sends a STACK to the VAL II controller. This action re-initializes the communication link.

4.3.3 VAL II Message Protocol

There are two key rules that must be observed when transmitting messages to and from the supervisory interface.

1. All supervisory communications are initiated by VAL II.
2. After every communication initiated by a logical unit, there must be a reply from the CEC to that logical unit before the logical unit can initiate another message.

Because of the first rule, a transmission from the supervisor system is not accepted unless an outstanding transmission from that logical unit has been acknowledged. VAL II is directed to initiate communication in one of two ways, depending on the logical unit involved. The communication with the monitor input and output logical unit (LUN 2) is initiated when the software switches NETWORK and SUPERVISORY are enabled. These switches are enabled by typing the switch ENABLE command, at the VAL II terminal. Once these

Table 5. Control Message and Data Message Sequence

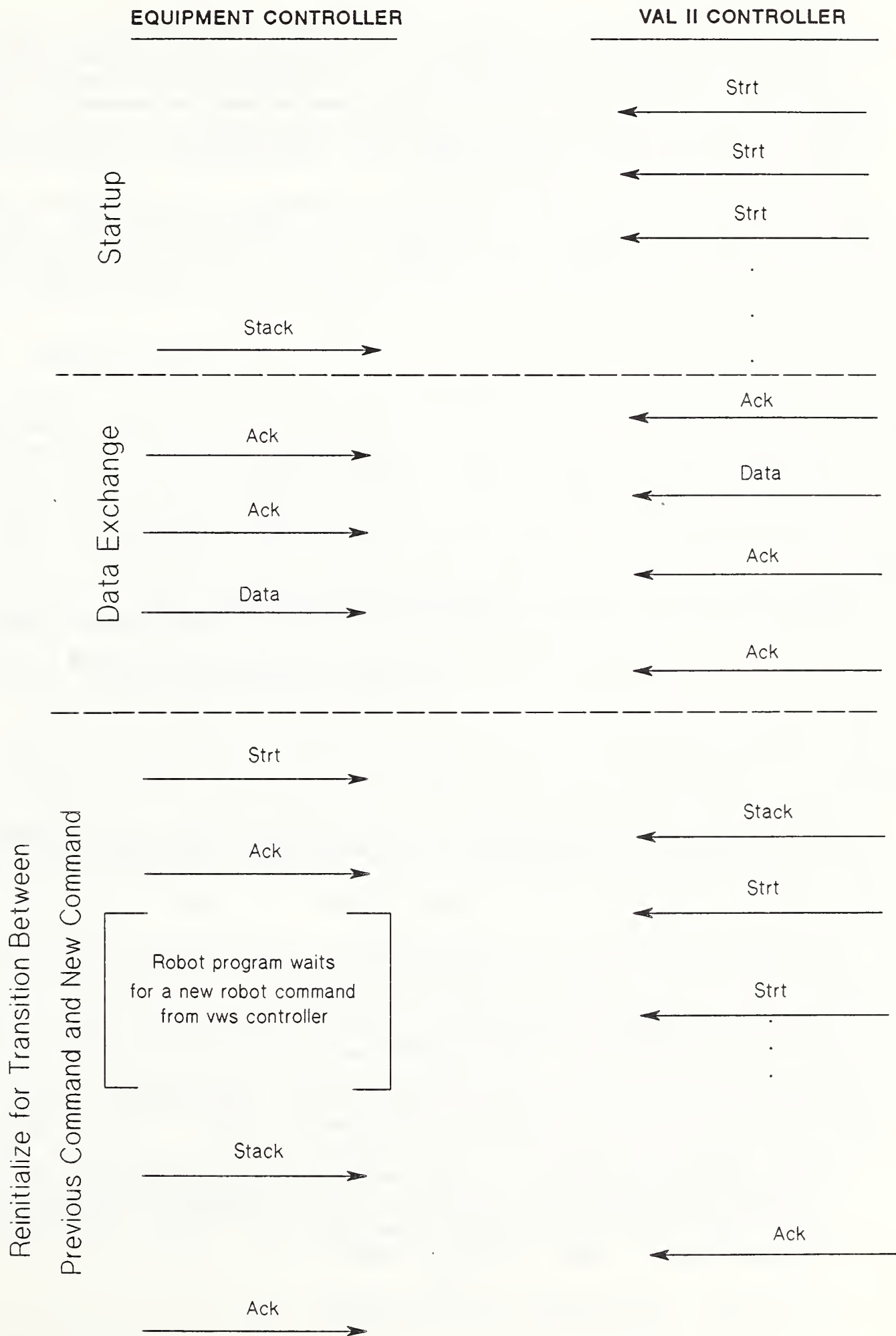
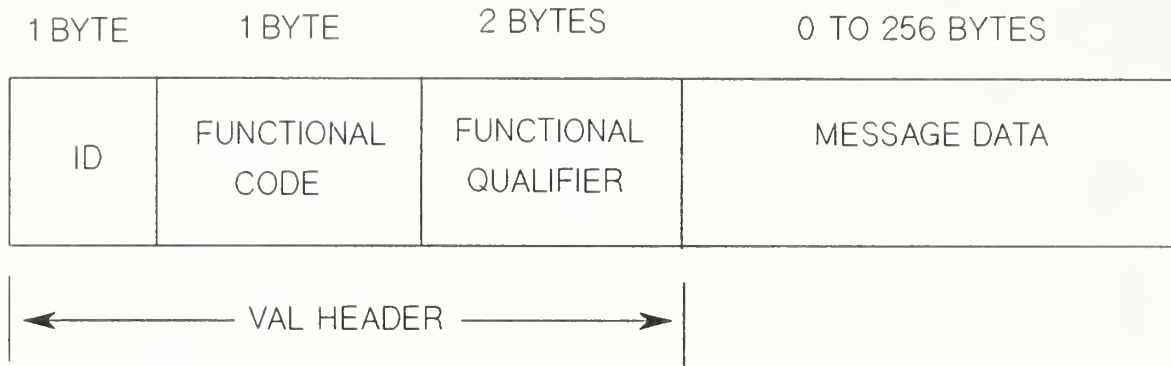


FIGURE 10. VAL DATA MESSAGE FORMAT



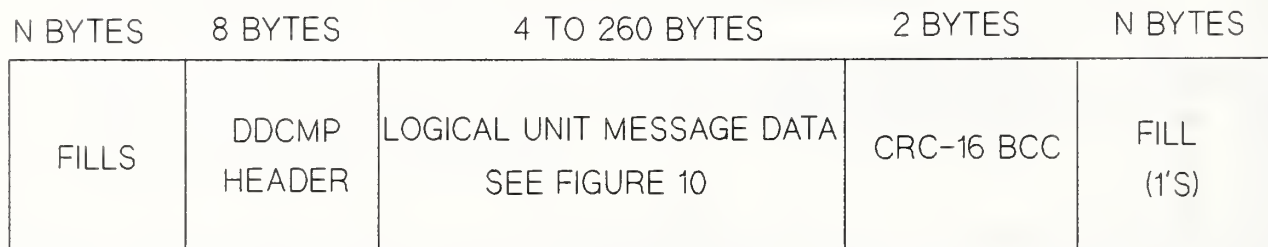
ID LOGICAL UNIT NUMBER, IT IS USED TO IDENTIFY THE LOGICAL UNIT ASSOCIATED WITH THE DATA MESSAGE

FUNCTIONAL CODE THIS CODE TELLS THE SUPERVISORY SYSTEM HOW TO RESPOND TO THE DATA MESSAGE

FUNCTIONAL QUALIFIER PROVIDES ADDITIONAL CONTROL INFORMATION

MESSAGE DATA THE ACTUAL MESSAGE BEING TRANSMITTED TO OR FROM THE VAL II CONTROLLER. WHEN THE TRANSMISSION IS FROM THE EQUIPMENT CONTROLLER TO THE VAL II CONTROLLER THE MESSAGE DATA FIELD IS A ROBOT COMMAND OR A ROBOT STATUS REQUEST

FIGURE 11. FORMAT OF TRANSMITTED MESSAGE



FILLS N FRAMING BYTES

DDCMP HEADER SAME AS SHOWN IN FIGURE 8

CRC-16 BCC BLOCK CHECK ON MESSAGE DATA

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

switches are enabled, the keyboard of the VAL II terminal is locked-out and no other commands can be entered at the terminal. After the start-up communication phase is completed, the LUN 2 transmits a message to elicit input of the next monitor command from the CEC. This is equivalent to the prompt normally outputted to the system terminal. For other logical units, communication is initiated when some input or output is requested of VAL II.

Once a logical unit has initiated a communication, rule number 2 must be observed. That is, the CEC must respond to the initial message before the logical unit can initiate another communication. This rule does not prevent other logical units from sending or receiving messages while another logical unit is waiting for its reply. [UNI2]

4.3.4 Message Format

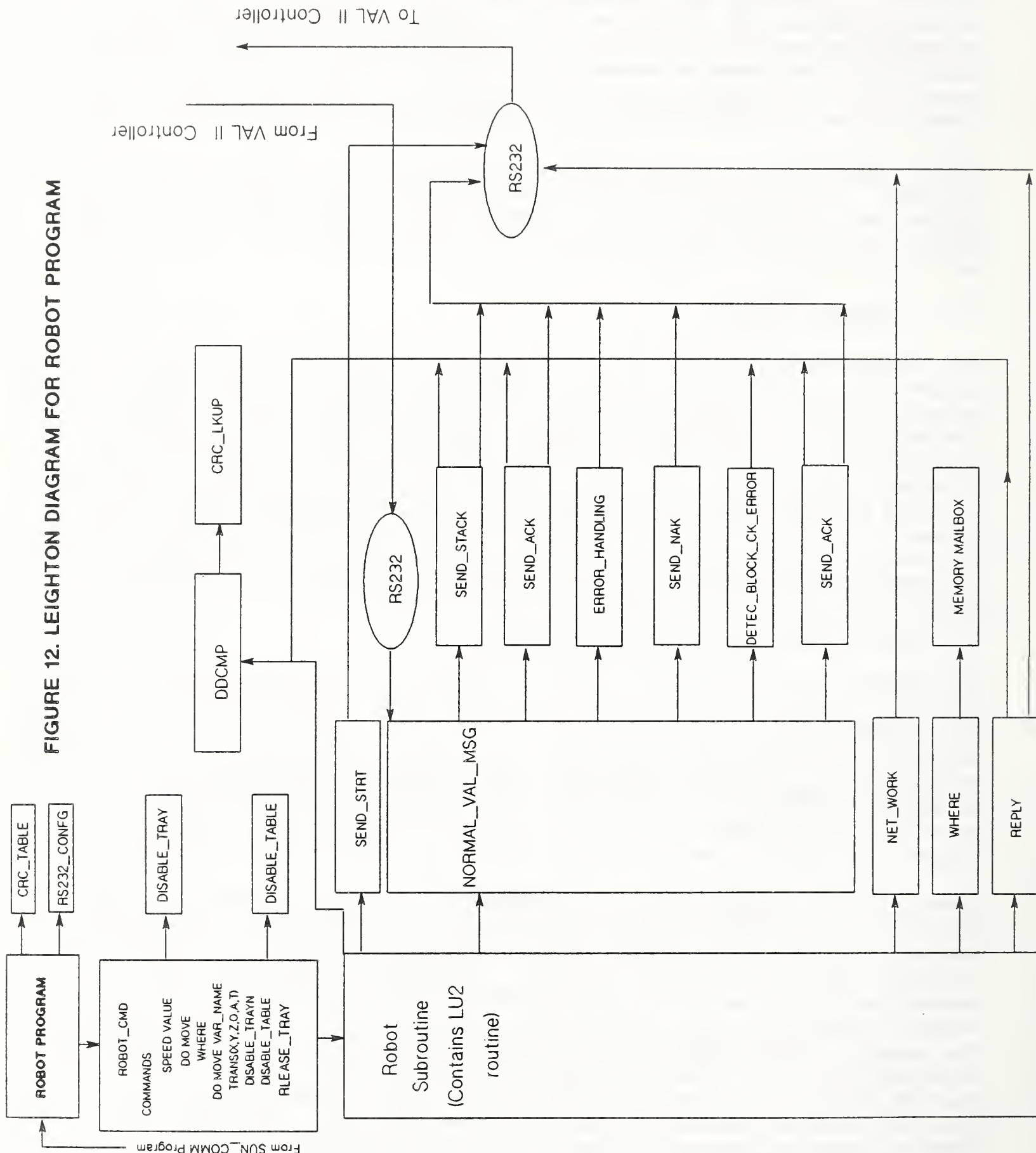
The formats of the message records sent and received at the top level of the supervisory communication system are shown in Figure 10. This format is used both for records transmitted by VAL II's Logical Units and for records transmitted by the supervisory system. Before the VAL II data message is sent over the physical link to the CEC, fill bytes and a DDCMP header are attached to the front of the data message and a BCC is attached to the end of the data message. The actual format sent over the physical link is shown in Figure 11. The DDCMP header consists of eight bytes which includes two bytes for the block check character (a CRC-16 type). Figure 8 shows the format of the DDCMP header, which contains all those fields up to the DATA field.

4.4 ROBOT COMMUNICATION SOFTWARE

The program for interfacing the CEC to the VAL II controller resides on the CEC. The communication protocols described above are also implemented in CEC's ROBOT program. This program resides in its own partition. It consists of a collection of subroutines that: (1) perform tasks that ensure safe robot operations, (2) a driver subroutine that controls communication with the supervisory system and (3) a collection of subroutines that perform tasks specified in the DDMCP. The scope of control, the hierarchy relation, and the external interfaces are shown in Figure 12 for the ROBOT program.

There are four safety interlock commands that activate subroutines which perform safety checks. These safety checks are performed to ensure that each mechanical system is not in an active state when the robot is commanded to interact with that mechanical system. For example, before the robot approaches the roller tables, the CEC locks the tray to prevent access by the material handling system. To perform this safety interlock, the CEC issues the command `DISABLE_TRAYN`. This prevents the material handling system from removing the tray. After the robot has finished placing parts on the tray, the CEC will issue a command `RELEASE_TRAYN`. This command unlocks the tray so that the material handling system can remove the tray. The CEC issues similar types of commands to lock the machine tool before the robot approaches the machine tool table and unlocks it after the robot departs.

FIGURE 12. LEIGHTON DIAGRAM FOR ROBOT PROGRAM



THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

Figure 13 shows a flowchart that includes a device-lockout check. This device-lockout is required to ensure that the machine table is not in motion when the robot is commanded to move from location "SAFE" to a work volume (WVN) such as the vise, located on the machine table. The device-lockout is used to implement safety interlocks, in both the ROBOT program and the MONARCH program.

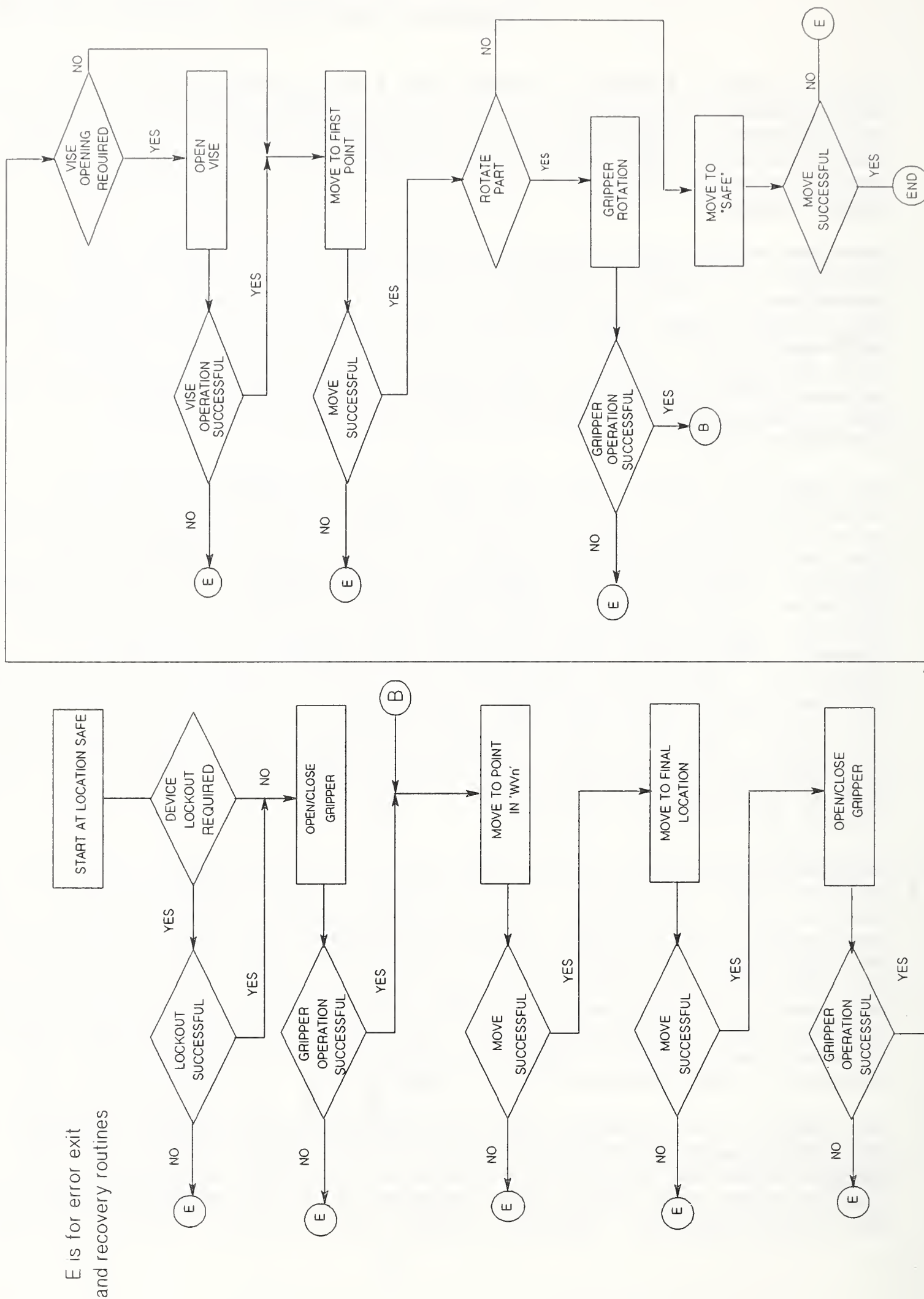
There are two important subroutines which implement the communication protocols for the VAL II supervisory system and for the DDCMP. These subroutines are "Robot" and Normal_val_msg. The Subroutine " Robot" receives VAL II data messages from subroutine Norm_val_msg. These data messages originated within the VAL II system at logical unit 2 and at logical unit 3. These Subroutines decode the VAL II data message, issue the appropriate response to the supervisory system, collect robot monitor commands, and send them to the appropriate subroutines for transmitting to the VAL II supervisory system.

The subroutine "Robot" serves the function of the communication driver between CEC and the VAL II supervisory system. Within this subroutine, there is a loop structure, which is called the LU2 routine. Figures 14 through 17 show the flowcharts for this routine. This LU2 routine interprets the monitored outputs which are transmitted during command execution and program execution. Such outputs include a message indicating the beginning of the execution for the command and a message indicating the completion of the command. These outputs are transmitted from the LUN 3 of the VAL II controller. Those robot instructions that are not monitor commands must be preceded by the reserved word DO. For example, to execute the MOVE command, the CEC prefixes a DO to the MOVE command before transmitting this command to the VAL II. LU2 parses the DO command and prevents another DO command from executing until the previous DO MOVE command has completed execution. Figure 12 shows that the LU2 routine calls the subroutine Network to transmit all commands to the VAL II controller.

Program initiation and program completion are also monitored by the LU2 routine. This monitoring process is described in a flow chart of the LU2 routine, shown in Figure 14. This monitoring process is performed by interpreting the first three bytes of VAL header, which precedes the data message transmitted from the LUN 3. These bytes, shown in Figures 10 and 14, are referred to as control bytes (CB). In particular, the message code, contained in the third byte of the VAL header is used to determine program initiation and program completion. The format of the data message from LUN 3 is the same as shown in Figure 10, except that the message code of LUN 3 replaces the functional qualifier in Figure 10.

The return of the requested status information from the VAL II controller is decoded and monitored by the LU2 routine of the "Robot" subroutine. For example, after the WC issues a status request, the LU2 routine calls the subroutine Net_work which in turn transmits the status request to LUN 2 of the VAL II controller. The LUN 2 returns the information requested by the

FIGURE 13: FLOWCHART FOR ROBOT MOVE FROM LOCATION "SAFE" TO WORK VOLUME LOCATIONS AND RETURN TO "SAFE" [REF: E. B. MAGRAB]



E is for error exit and recovery routines

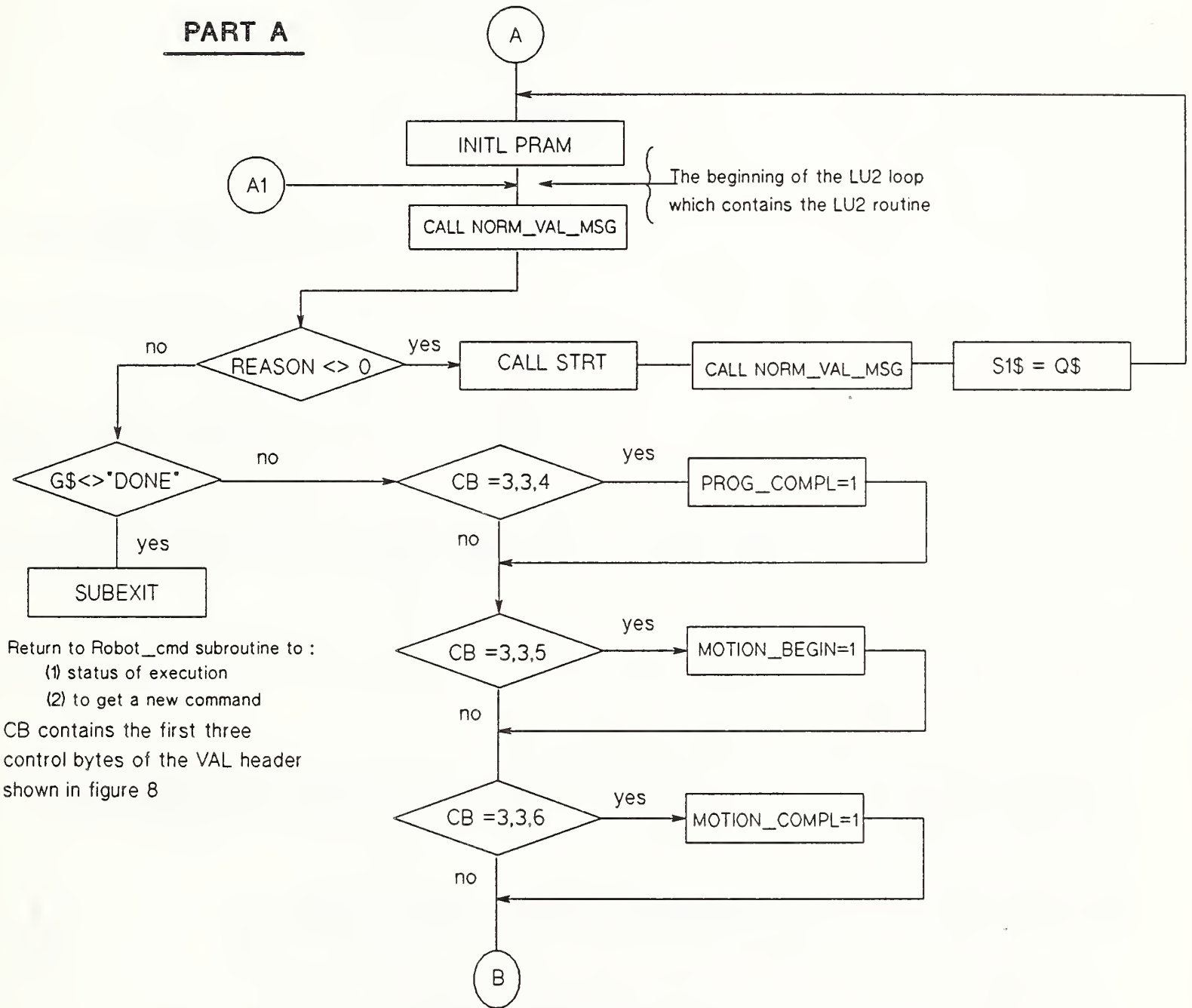


FIGURE 14 : Flowchart for Subroutine Robot:(part A LU2 Routine)

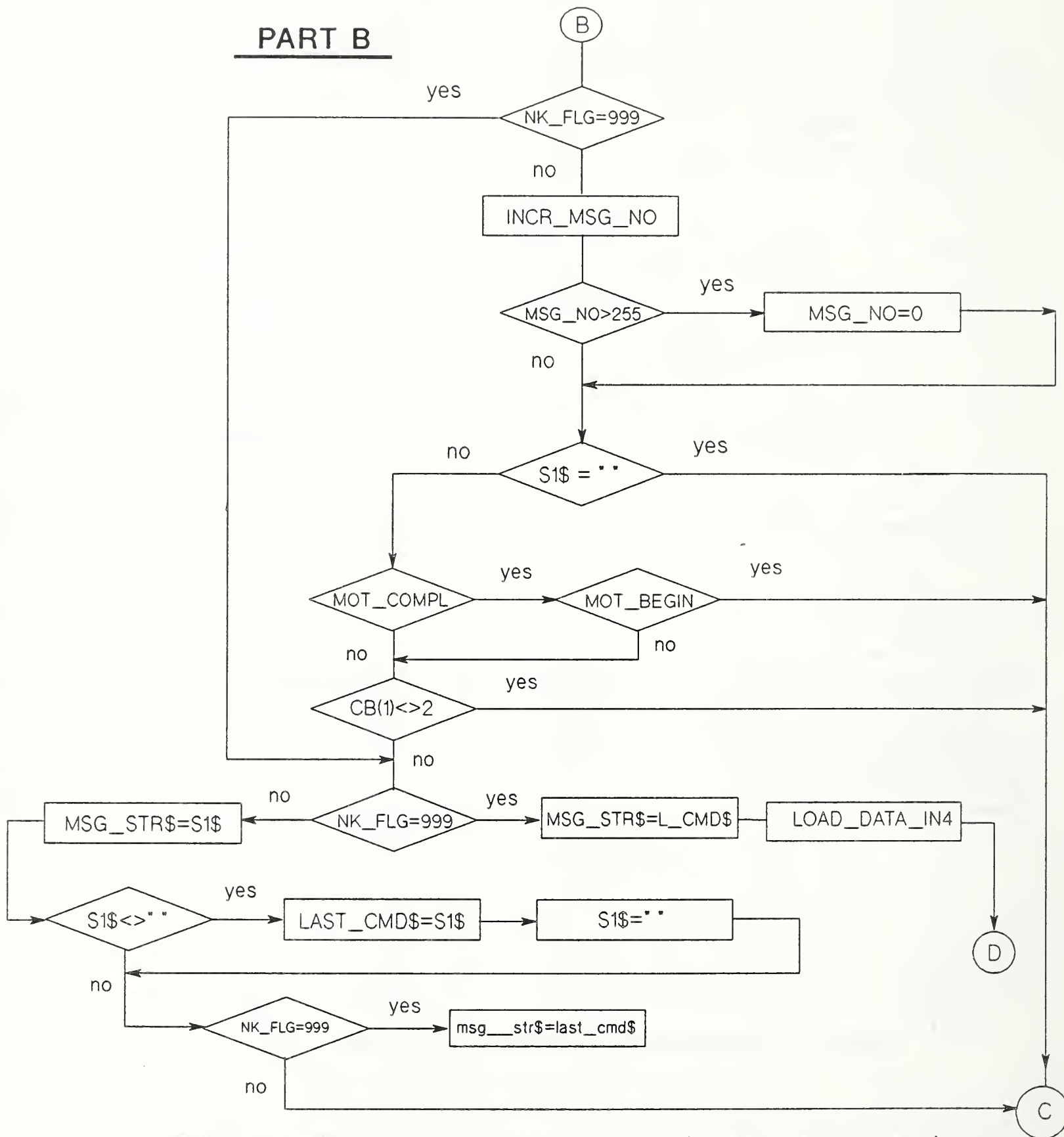


FIGURE 15 : Flowchart for Subroutine Robot:(part B of LU2 routine)

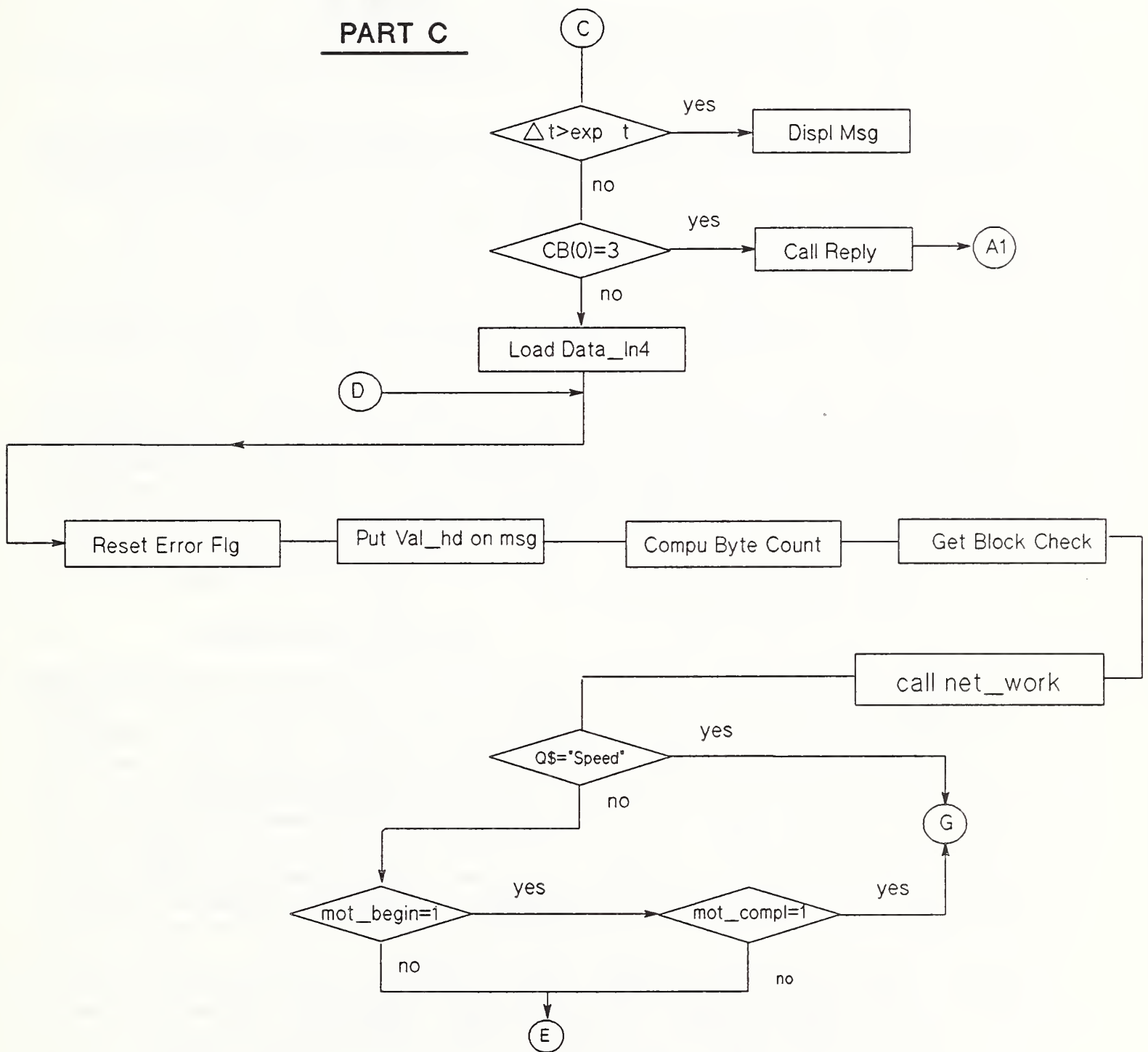


FIGURE 16 : Flowchart for Subroutine Robot:(part C of LU2 routine)

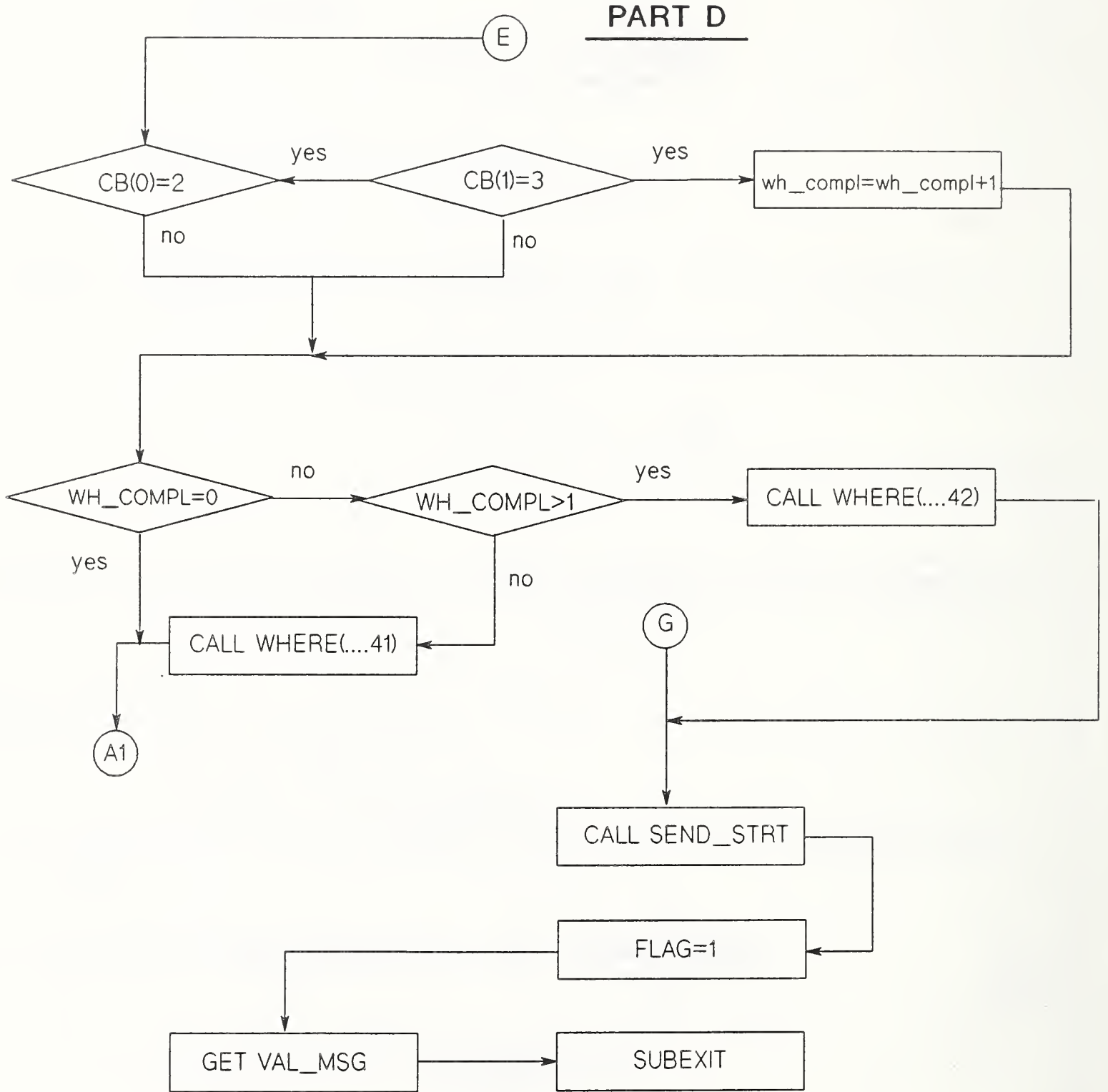


FIGURE 17: Flowchart For Subroutine Robot,(part D,End of LU2 routine)

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

status command. In the case of a WHERE status request, the LU2 routine receives from the VAL II the transmitted location data consisting of both world coordinates and joint coordinates. During this transmission, the LU2 routine will not accept a new command.

If any errors are caused by hardware malfunction, the LUN 3 will report the error message within the message data field, shown in Figure 10. The field length of message data is 256 bytes, starting at byte 5.

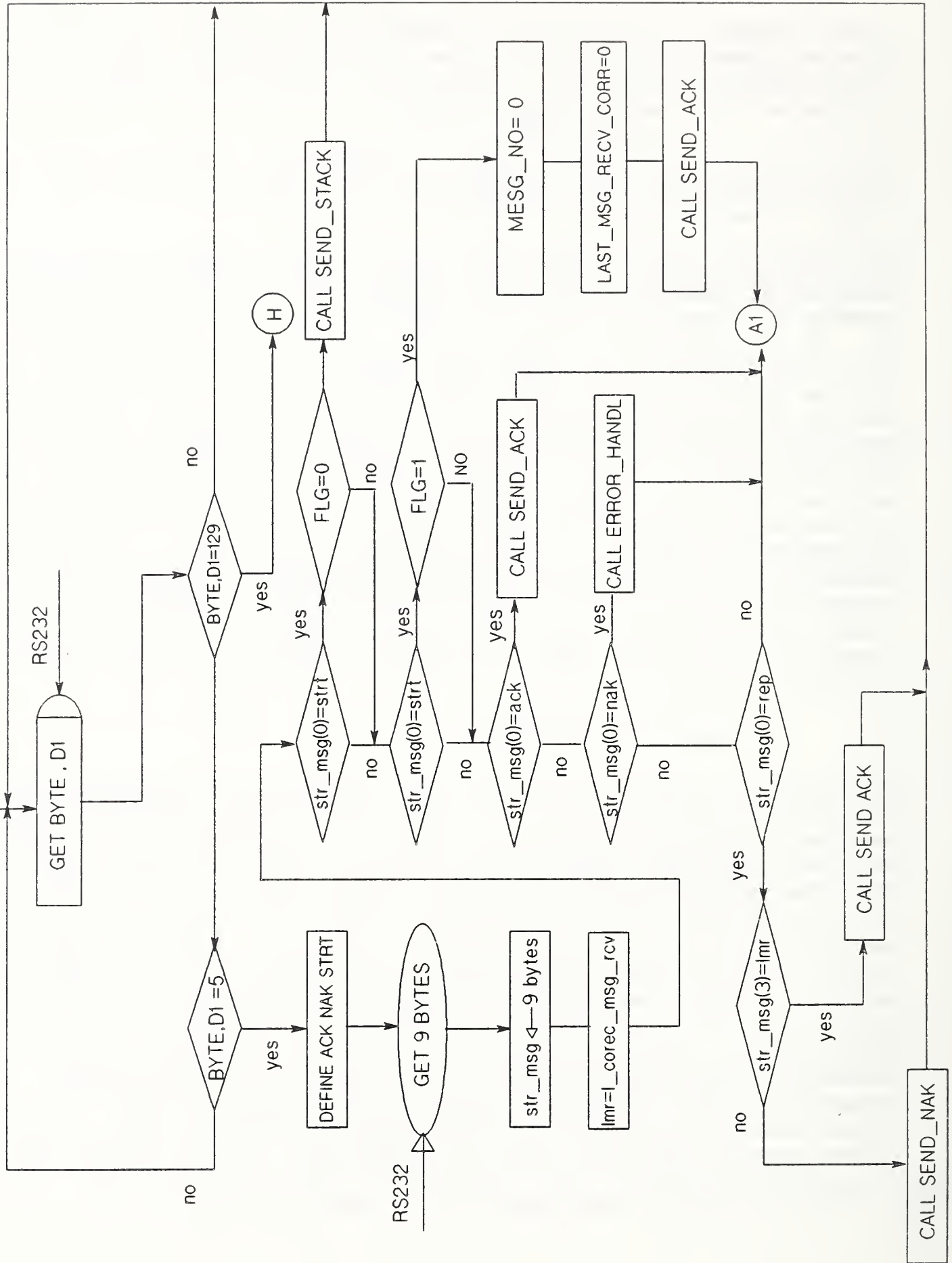
The subroutine Norm_val_msg collects all incoming messages from the VAL II supervisory system and determines whether the message is a DDMCP control type message as shown in Figure 9 or a VAL II data type message as shown in Figure 10. If the message is a DDMCP control message type, this subroutine branches to an ENQ routine to determine what type of response is required. This ENQ routine calls the appropriate subroutine which sends the required response to the VAL II supervisory system. The control logic for this ENQ routine is displayed by the flow chart shown in Figure 18. However, if the message is a VAL data message, it will branch to a SOH routine to retrieve the remaining portion of the data message and then it sends the data message to the Robot subroutine for subsequent action. This subsequent action is usually one of two types: (1) issuing a new command and (2) replying to a previous VAL II message. The control logic for this SOH routine is shown by the flowchart in Figure 19. The overall influence for the "NORMAL_VAL_MSG" subroutine and the "ROBOT" subroutine is shown the Leighton diagram of Figure 12 and the detail control logic for these two subroutines is displayed by the flowcharts shown in Figs. 14 through 19.

5.0 HYDRAULIC CONTROLLER

The hydraulic controller is an HP-9836 computer, with 720,886 bytes of Memory. The communication link between this controller and the central equipment controller (CEC) is the HPIB bus. The hydraulic controller receives individual commands from the CEC. After receiving commands from the CEC, the hydraulic controller sends an appropriate set of commands to the data acquisition unit #2 via the HPIB line. The data acquisition unit closes switches that cause hydraulic valves to operate. The operations of these valves activate clamps which secure the workpiece to the machine table. Other commands issued from the CEC to the hydraulic controller cause the data acquisition unit #2 to read the X-position and the Y-position microswitches of the pallet. These microswitches determine whether the pallet is registered against the X-position stops and the Y-position stops. Other commands from the CEC to the hydraulic controller cause the level of the forces exerted by the pallet swing clamps to be read. The reading of these forces ensures that the pallet is sufficiently clamped to the machine table.

FIGURE 18. ENQ ROUTINE OF SUBROUTINE Norm_val_msg FOR RETRIEVING AND MONITORING CONTROL MESSAGES

ENQ ROUTINE



SOH ROUTINE

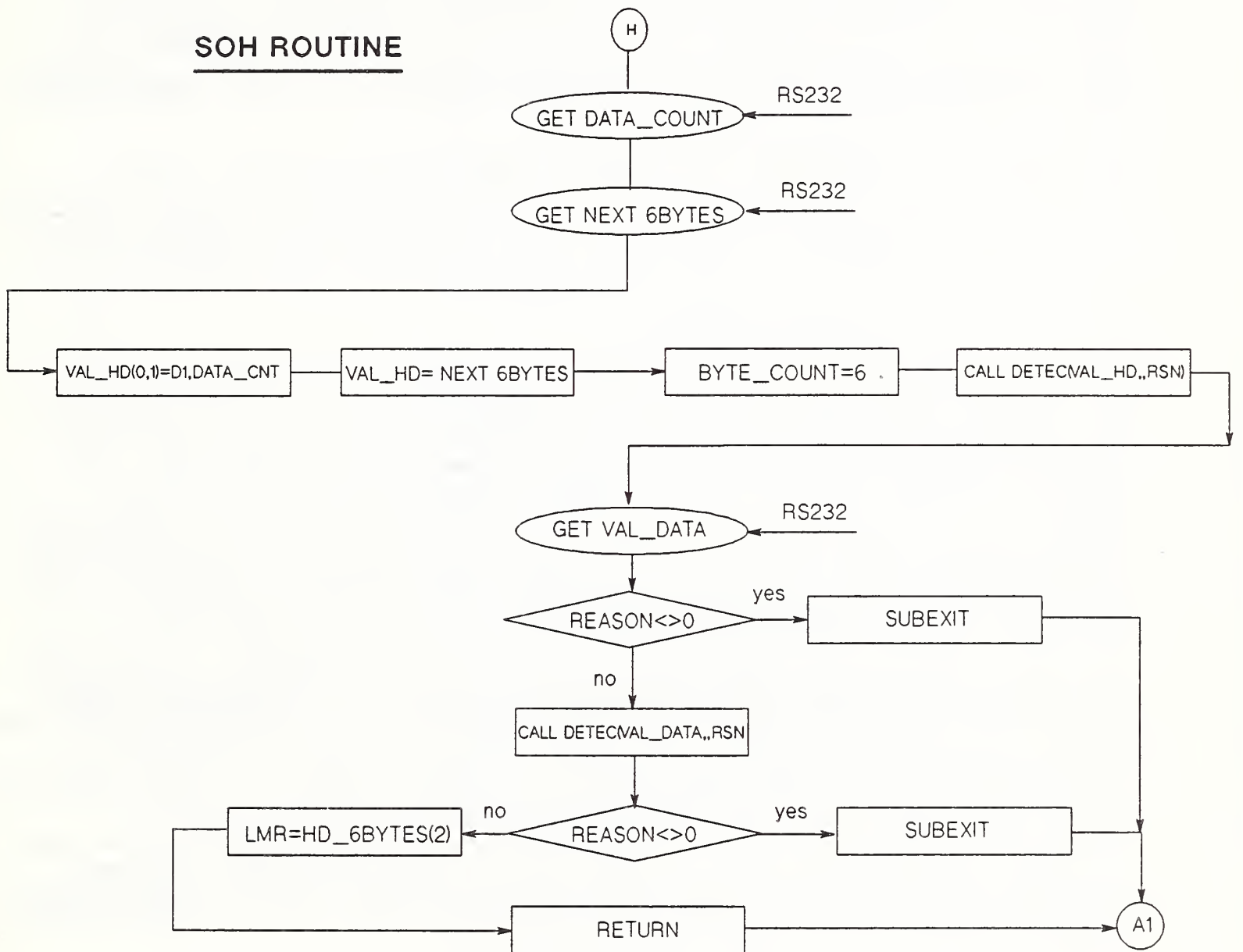


FIGURE 19. SOH ROUTINE OF SUBROUTINE Norm_val_msg FOR RETRIEVING DATA MESSAGES

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

5.1 MAJOR SOFTWARE FEATURES

The hydraulic controller software consists of three initialization subroutines and one task manager subroutine. Two of the initialization subroutines are used to set up communication over the HPIB link. One of these initialization subroutines establishes communication with the Central Equipment Controller and the other initialization subroutine establishes communication with the data acquisition unit #2. The third initialization routine performs the task of depressurising the hydraulic valves to ensure that the swing clamps are in the fully released position before sending a new command to activate these clamps.

The task manager function is performed by subroutine "Options". This subroutine operates in a continuous loop to receive the next command from the CEC. After it receives a new command, this subroutine parses this command to determine which subroutine it should call to carry out this command.

6.0 Concluding Observations

In a multi-controller and multi-program environment it is essential that well-defined handshaking protocols are implemented and strictly followed. The accuracy of the message being transmitted between controllers must be ensured by CRC-16 block-check characters or some other suitable BCC. The concept of having a CEC to serve as master to all other controllers is essential for proper program synchronization and for the proper scheduling of command driven tasks. The driver program for the CEC should be capable of parsing an incoming command set and routing the individual commands to other controllers, which carry out the intended function of the individual commands. Hardware handshaking is essential. It is not enough for the CEC to issue a command and assume that it is executed successfully. There must be positive verification by feedback from the process to ensure that the command was executed successfully. This is hardware handshaking. In the VWS configuration, this positive verification takes the form of sensing the voltage levels across microswitches and reading the output of strain gages to ensure, for example, that the clamping force exerted by the vise is sufficient to keep the workpiece from moving. Good programming methods are essential. One should strive to implement structural programs, designing each subroutine to handle a specific task or a closely related group of small tasks.

Error trapping and error recovery is a programming area that requires a great deal of thought and planning. Instant display of error messages is helpful for diagnosing the cause of errors and is essential for implementing an orderly recovery plan.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

APPENDIX I

EQUIPMENT CONTROLLER'S COMMAND SETS
REF [MAG1]

A. MACHINE TOOL COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
CYCLE_START	Starts the execution of the program chosen with the "PROGRAM SELECT" command after first setting the table interlock flag to prevent the vise from being opened or the robot from approaching the table.
RELEASE_TABLE	A software interlock command that permits the vise to be opened or the robot to approach the table.
STATUS	Tells whether or not the machine tool is executing
DOWNLOAD_HP	Transfers an NC program from the workstation controller to an equipment controller memory file. Any previously transferred file will be overwritten.
DOWNLOAD_GE	Transfers an NC program residing in the equipment controller's memory file to the machine tool controller. Transfer can be done any time prior to another "DOWNLOAD_HP" command.
UPLOAD_GE(NNNNNN.PRO or NNNNNN.GSU)	Transfers either an NC program with program number NNNNNN.PRO or a subroutine with subroutine number NNNNNN.GSU from the machine tool controller to an equipment controller memory file. Execution of this command overwrites any previously transferred file.
UPLOAD_SUN	Transfers the NC program residing in an equipment controller memory file to the workstation controller. This command can be done any time prior to another "UPLOAD_GE" command.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

TABLE (L1/L2/.../Ln)	Transfers the contents of the tool length offset table to the equipment controller and then selects the appropriate numerical values for transfer to the workstation controller. The L_j , $j=1,2,\dots,n \leq 8$ are the line numbers in the table containing the numerical values of interest. They must appear in ascending order: $L_1 < L_2 < \dots < L_n$.
PROGRAM_SELECT(NNNNNN)	Selects the program with program number NNNNNN to be executed from the machine tool controller's memory with the next "CYCLE_START" command.
PROGRAM_DELETE(NNNNNN.PRO)	Deletes the program with program number or NNNNNN.GSU) NNNNNN.PRO or the subroutine with subroutine number NNNNNN.GSU from the machine tool controller's memory.

B. ROBOT COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
WHERE	Obtains the current location of the robot in world coordinates and in terms of its joint angles.
SPEED N	Specifies the speed of all subsequent robot moves.
DO APPRO locat/dist	Moves in joint interpolation mode to the point 'locat' along the Z-axis, offset from 'locat' an amount 'dist' mm.
DO APPROS locat/dist	Moves in a straight line to the point 'locat' along the Z-axis, offset from 'locat' an amount 'dist' mm.
DO DEPART dist	Moves back from the current point along the Z-axis 'dist' mm in joint interpolation mode.
DO DEPARTS dist	Moves back from the current point along the Z-axis 'dist' mm in a straight line.
DO MOVE var_name	Moves robot to location 'var_name' using joint interpolated motion.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

DO MOVE var_name Moves robot to location 'var_name' using straight line motion.

DO MOVE var_name:TRANS (X₀/Y₀/Z₀/90/-90/T

The point 'var_name' defines a local coordinate system (X,Y,Z) located in the bottom plane of the robot's 6th axis that has a special angular orientation (90°, -90°, T), where T is the rotation, in degrees, about the local Z-axis. This command moves the robot to a new point (X₀, Y₀, Z₀) relative to the local coordinate system (X,Y,Z) defined at 'var_name'.

DISABLE_TRAYN A software interlock command that locks out any movement of tray N (N=1 or 2) until released with a "RELEASE_TRAYN" command.

RELEASE_TRAYN A software interlock command that permits movement of tray N (N=1 or 2).

DISABLE_TABLE A software interlock command that locks out any movement of the machine tool table until released with a "RELEASE_TABLE" command.

RELEASE_TABLE A software interlock command that permits the machine tool controller to move its table; that is, execute a program with the "CYCLE_START" command.

EXECUTE name Execute program 'name' residing in the robot controller's memory. Used to run a special process control program for real-time path control.

C. VACUUM COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
ON	Turns vacuum on
OFF	Turns vacuum off

D. GRIPPER COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
OPEN	Opens gripper jaws the maximum amount

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

CLOSE Closes gripper jaws until it stops against an object or against its own fingers (empty).

ROTATE_N Rotate indexer N (N=0°, 90°, 180° or 270°)

E. VISE COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
OPEN	Opens vise's jaws.
CLOSE	Closes vise's jaws.
DISABLE_TABLE	A software interlock command that locks out any movement of the machine tool table until released with a "RELEASE TABLE" command.
RELEASE_TABLE	A software interlock command that permits the machine tool controller to move its table; that is, execute a program with the "CYCLE_START" command.

F. HYDRAULIC COMMAND SET

<u>COMMAND</u>	<u>EXPLANATION</u>
CLAMP_PALLET	Sequentially actuates two pallet positioning rams and two swivel-rocker clamps.
RELEASE_PALLET	Sequentially retracts the devices actuated with CLAMP_PALLET.

G. STATUS COMMAND SET

<u>COMMAND</u>	<u>REPLY</u>
GRIPPER	GRIPPER IS OPEN and INDEXER ROTATED N deg or FINGERS ARE d mm APART
TABLE	TABLE IN PARK POSITION or TABLE NOT IN PART POSITION
WISE	WISE JAWS ARE d mm APART and VISE IS LOCKED PART IN VISE or WISE IS UNLOCKED PART NOT IN VISE

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

TRAYS TRAY #1 IN POSITION TRAY #2 IN POSITION
 or and or
 TRAY #1 NOT IN POSITION TRAY #2 NOT IN POSITION

VACUUM VACUUM IS ON
 or
 VACUUM IS OFF

TLI_TABLE N/M₁/M₂/.../M_N N ≤ 8

where M_j, j=1,2,...,N, are floating point numbers.

CHANGER_1 CHANGER_1 FINGERS DID NOT RELEASE
 or
 CHANGER_1 FINGERS RELEASED

MONARCH ACTIVE
 or
 IDLE

WHERE_ROBOT N₁ ,/N₂/.../N₁₂

where the first six floating point numbers are the robot's current world coordinates (x,y,z,θ,A,T) and the last six numbers are the robot's joint angles in degrees, except for the third number of this group which is the extension of the boom arm in millimeters.

PALLET PALLET CLAMPED
 or
 PALLET NOT CLAMPED

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

REFERENCES

[ANSI X 3.28-86]

Procedures for Use of Communication Characters of the American National Standard Code of Information Exchange in Specifying in Data Communication, 1976 Rev 1986.

[DEC1]

Digital Equipment Corporation; DECNET Digital Network Architecture, Digital Data Communication Message Protocol (DDCMP), specification version 4.0"; March 1978.

[EIA RS 491]

Interface Between a Numerical Control Unit and Peripheral Equipment Employing Asynchronous Binary Data Exchange Over Circuits Having RS-423-A Electronical Characteristics (from EIA Standards Proposal No. 1292-A formulated under the cognizance of EIA IE-31 Committee on Numerical Control Systems and Equipment), 1982.

[GEK1]

General Electric Company; "DNC/FMS Communication Manual, GEK 25348"; June 1984.

[HP 1]

Hewlett-Packard Company; "Basic Software Configuration Manual for the HP 9000 series 500/600/700/ Model 20, part no 97050-90045"; September 1983.

[KRA1]

Kramer, Thomas R.; "Process Plan Expression, Generation, and Enhancement for the Vertical Workstation Milling Machine in the Automated Manufacturing Research Facility at the National Bureau of Standards"; NBSIR 87-3678; November 19, 1987.

[KRA2]

Kramer, Thomas R.; "The Graphics Subsystem of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards"; to be published as an NBSIR, 1988.

[KRA3]

Kramer, Thomas R.; " The vws_cadam User Interface of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards"; NBSIR 88-3738; March 10, 1988.

[KRA4]

Kramer, Thomas R.; The Data Execution Module of the Vertical Workstation of the AMRF at the National Bureau of Standards; NBSIR 88-3704; January 6, 1988.

THE VERTICAL WORKSTATION'S EQUIPMENT CONTROLLERS

[K&J1]

Kramer, Thomas R.; and Jun, J.; "Software for an Automated Machining Workstation"; Proceedings of the 1986 International Machine Tool Technical Conference; National Machine Tool Builders Association, Chicago, Illinois; September, 1986; pp. 12-9 through 12-44.

[K&J1]

Kramer, Thomas R.; and Jun, J.; "The Design Protocol, Part Design Editor, and Geometry Library of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards"; NBSIR 88-3717; January 28, 1988.

[MAG1]

Magrab, Edward B.; "Vertical Machining Workstation of the AMRF: Equipment Integration"; Proceedings of the ASME Symposium on Intelligent and Integrated Manufacturing, Chicago, Illinois; December, 1986.

[NA&J]

Nakpalohpo, Ibrahim; and Jun, JAU-Shi; "Automated Equipment Program Generator and Execution System of the AMRF Vertical Workstation"; available from the authors; 17 pages.

[RUD1]

Rudder, Fred.; "Operation Manual for VWS Equipment Controllers", to be published as an NBSIR, 1988.

[UNI1]

Unimation Inc.; "4070 Series Equipment Manual for VAL II and VAL PLUS Operating System 398AB1"; May, 1986.

[UNI2]

Unimation Inc.; "Program Manual, User's Guide to VAL II, Version 2.0, 398AG1"; February, 1986.



READER COMMENT FORM

Document Title The Vertical Workstation's Equipment Controllers

This document is one in a series of publications which document research done at the National Bureau of Standards' Automated Manufacturing Research Facility from 1981 through March, 1987.

You may use this form to comment on the technical content or organization of this document or to contribute suggested editorial changes.

Comments: _____

If you wish a reply, give your name, company, and complete mailing address: _____

What is your occupation? _____

NOTE: This form may not be used to order additional copies of this document or other documents in the series. Copies of AMRF documents are available from NTIS.

Please mail your comments to: AMRF Program Manager
National Bureau of Standards
Building 220, Room B-111
Gaithersburg, MD 20899



U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET <i>(See instructions)</i>	1. PUBLICATION OR REPORT NO. NBSIR 88-3769	2. Performing Organ. Report No.	3. Publication Date APRIL 1988
4. TITLE AND SUBTITLE The Vertical Workstation's Equipment Controllers			
5. AUTHOR(S) C. Denver Lovett			
6. PERFORMING ORGANIZATION <i>(If joint or other than NBS, see instructions)</i> NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		7. Contract/Grant No.	8. Type of Report & Period Covered
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS <i>(Street, City, State, ZIP)</i>			
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.			
11. ABSTRACT <i>(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)</i> This document describes how the central equipment controller (CEC) of the Vertical Workstation (VWS) operates as an interfacing controller between an upper-level controller, the workstation controller (WC), and three lower-level controllers and two data acquisition units. The CEC usually acts as a master controller when communicating with the three lower-level controllers and as a slave controller when communicating with the WC. The three lower-level controllers are computer-based. One of these is a robot controller, one is a machine tool controller and the third is a general purpose computer operating as a hydraulic controller. Each controller operates under a different communication protocol. The robot controller operates in a monitor command mode, the machine tool controller operates in a DNC mode (with NC program transfers) and the hydraulic controller operates in a command-driven program mode. The sensor inputs for achieving hardware handshaking are retrieved from the two data acquisition controllers. At the heart of the CEC is a collection of eight programs that are used to coordinate the activities of the three lower-level controllers and devices such as the vacuum and the NBS-designed robot gripper. The CEC operates along with the WC, in a multi-controller and multi-programming environment, to achieve a high degree of machining flexibility by employing VWS features such as: (1) an automated NC-code generator, (2) an automated process planning system & (3) an automated robot programming generator.			
12. KEY WORDS <i>(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)</i> Communication Protocols; Computer-aided manufacturing; Computer Interface; Direct Numerical control; Equipment controllers; Hardware Handshaking; Machine Tool; Manufacturing; NC Code; Programmable, Flexible Automation; Robot;			
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161		14. NO. OF PRINTED PAGES 67	15. Price \$13.95



