ALLEN-BRADLEY
A Rockwell International Company

User's Manual

Series 8400 PAL (Programmable Application Logic)

Price: $50.00

Important User Information

Because of the variety of uses for the solid state equipment described herein, and because of the differences between it and electromechanical equipment, you must satisfy yourself as to its acceptability for each of your applications. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages that may result from the installation or use of this equipment.

The illustrations, charts, and layout examples shown in this manual are intended solely to help you understand the text, not to guarantee operation. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company will not assume responsibility for actual use based upon illustrations of applications.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment or software described in this text.

Table of Contents

Table of Contents

1.0 Chapter Overview

Read this chapter before you begin using this manual. This chapter tells you how the manual is organized and provides you with PAL background information.

1.1 The Purpose of This Manual

This manual will show you how to produce a PAL program in ladder diagram form and load the program into an Allen-Bradley Series 8400 control. It will also show you how to edit a PAL program on the Kaypro computer.

1.2 Audience

We assume that you have PAL programming experience but that you are not familiar with the Allen-Bradley Series 8400 numerical control. We've directed this manual towards the machine tool or robot builder that is integrating a Series 8400 control with a machine tool or robot.

1.3 Manual Contents

This manual contains five chapters.

- Chapter 1 - This Manual
- Chapter 2 - 8400 PAL Overview
- Chapter 3 - Programming Guidelines and Examples
- chapter 4 - Using PAL.COM
- Chapter 5 - 8400 PAL Instructions

If this is your first experience with 8400 PAL, we recommend that you pay special attention to chapters 2 through 4. Before you can program 8400 PAL effectively, you'll have to be familiar with the background information contained in these chapters.

Chapter 5 Contains definitions of instructions. You will need them to develop your 8400 PAL programs.

We've tried to be as specific as possible in giving you this information. There are always variations in applying the theory. It's a good idea to practice, safely, with the information offered in this manual, and develop your own way of writing PAL. Always verify the correct and safe operation of your PAL program and the equipment it controls.

If you find any errors or inadequacies in this manual, please notify the CNC Technical Writing Department of Commercial Services, Industrial Computer Group, Allen-Bradley Company, Highland Heights, Ohio.

1.4 Related Publications

You can find information relating to specific questions on PAL programming and applications in these publications.

Publication Title	Publication Number	Part Number
8400 GLC PAL Flags	8400-4.2.1	929295-01
8400 Robot PAL Flags	8400-4.2.4	929236-01
8400 GP PAL Flags	8400-4.2.5	929245-01
Series A 8400 Technical Information Manual	8400-4.3	929161-03
8400 GLC Operation and Job Entry Manual	8400-5.1.1	929158-02
8400 AMP Users Manual (Revision D Firmware)	8400-5.1.3	929206-02
8400 Robot Users Manual	8400-5.1.4	929169-02
8400 GP Operators Manual	8400-5.1.5	929169-02
8400 GLC Programming Manual (6.0)	8400-5.2.2	929170-02
8400 GP Programming Manual	8400-5.2.4	929171-02
8400 GLC AMP Users	8400-6.1.1	929174-01
8400 R Robot AMP Programmers Manual	8400 6.1.4	929208-01
8400 GP AMP Programmers Manual	8400-6.1.6	929207-02
Series B 8400B MP Installation Manual	8420-4.1.3	929286-01
Series B 8400 MP D	8420-5.1.3	929284-01

Bandit III CNC Installation Manual	BDT3-4.3	929155-01
Bandit III Users Manual	BDT3-5.1.3	929200-02
Bandit IV Installation Manual	BDT4-4.1	929285-01
Series B 8400 MP Bandit IV Standard PAL Data Sheet	BDT4-4.2.10	929293-01
Series B 8400 MP Bandit IV PAL Flags	BDT4-4.2.3	929297-01
Bandit IV D Users Manual	BDT4-5.1.3	929283-01

1.5 Terms And Conventions

The following list of terms and conventions makes the manual easier for you to read and understand.

We refer to the:

- Adjustable Machine Parameters as "AMP"
- Kaypro 2 and Kaypro 2X as "the computer"
- 2716 Eraseable Programmable Read Only Memory chip as "EPROM"
- the program that actually commands the machine to move the spindle and the axes creating a part is the "task program"
- Bay Technical Associates Model #953B EPROM Programmer as "the EPROM programmer"
- Allen-Bradley 8400 Custom Numerical Control as "the control"
- floppy diskette drive as "the drive"
- floppy diskette as "the disk"

Throughout the text of this manual you may see paragraphs beginning with the following words. These paragraphs have specific meanings:

- **Important:** Points out something important to the reader.
- **Caution:** Points out a situation with potential hazard to equipment.
- **Warning:** Points out a situation with potential hazard to equipment and/or personnel.

Pay close attention to these paragraphs.

Keys on the computer keyboard are shown in brackets in this text. For example, the return key is shown as [RETURN], and the 1 key is shown as [1].

1.6 Using This Manual

This manual shows you how to use the PAL program to create, modify and implement PAL.

Each chapter in this manual corresponds to a particular procedure in the development of your PAL ladder. This manual leads you through that procedure, step by step. You should familiarize yourself with this manual before you use the PAL ladder development program.

1.7 Chapter Summary

This chapter introduced you to to this manual and provided information you should be aware of before you begin. In the next chapter we introduce the PAL ladder diagram and how it communicates with the control and the machine.

2.0 PAL Communications

The PAL ladder diagram performs three activities when it controls machine functions:

- sequence machine functions
- operate machine functions
- monitor machine functions

PAL Communicates with the CNC executive software and descrete I/O to coordinate machine functions. See the block diagram in figure 2-1 on page 2-2.

The CNC executive sends commands to the PAL ladder diagram through H flags and G variables. The PAL ladder diagram sends information back to the CNC through N flags and L variables. The PAL ladder diagram also turns on preloaded messages stored in PAL through M flags.

The PAL ladder diagram communicates with the discrete I/O through R flags and V variables.

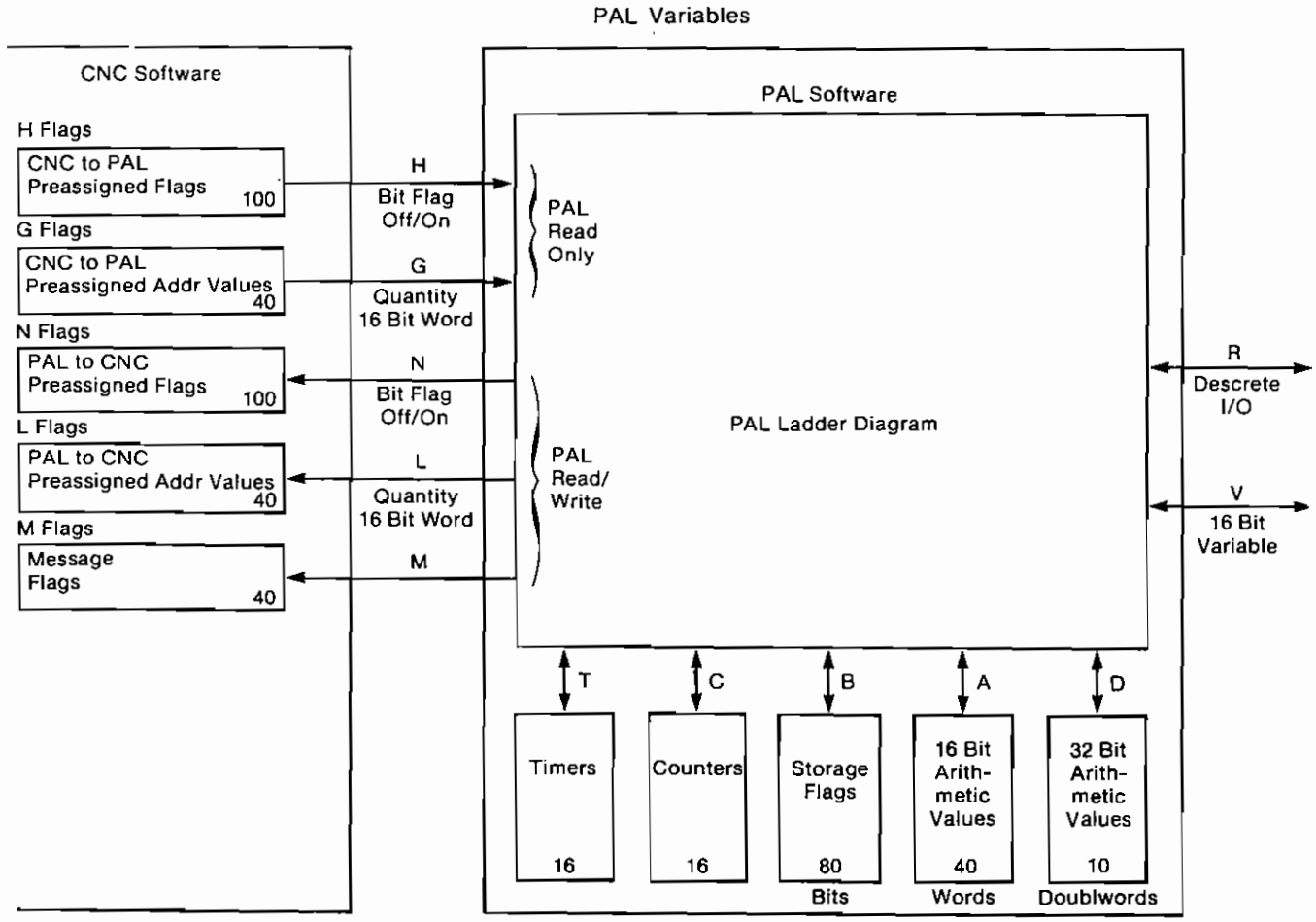
2.1 Flags and Variables

Flags are single bits of information. Variables are groups of bits that form a word (a 16 bit word or a 32 bit double word). Flags and variables may act on activities in PAL, in the CNC executive or in the machine.

The CNC executive and PAL ladder use flags and variables to:

- request that an activity take place
- monitor an activity that is happening or has happened
- cause a condition in an activity, such as inhibiting axis motion until the spindle is up to speed.

Figure 2-1
Communication
Block Diagram



H FLAGS 001-100

An H flag is a bit (on or off) that is sent from the CNC executive to PAL. In PAL an H flag is Read only. PAL cannot write (reply) back to the CNC using an H flag. PAL uses an N flag to send bit information to the CNC executive. H flags and N flags form a loop from the CNC to the PAL ladder and back.

The following list gives examples of the information that H flags carry to PAL and describe the mode of CNC operation.

Important: This is not a comprehensive list of H flags. It is only an example of the types of flags used in the 8400 PAL. Check your PAL Flags and Variables manual to find the H flags for your application.

--- LATHE INDEX ---

H FLAGS - READ ONLY

ADDRESS	NAME	MNEMONIC	MEMORY PAGE
H001	Power up	PORUP	103
H002	Axis 1 In Position	AX1IP	104
H003	Axis 2 In Position	AX2IP	105
H004	Axis 3 In Position	AX3IP	106
H005	Axis 4 In Position	AX4IP	107
H016	Axis 7 Move Request	AX7MV	118
H017	Axis 8 Move Request	AX8MV	119
H018	Axis 1 At Home Position	AX1HM	120
H019	Axis 2 At Home Position	AX2HM	121
H020	Axis 3 At Home Position	AX3HM	122
H026	K1 Relay Enable	RELEN	128
H027	Dry Run Mode	DRYRN	129
H028	Test Run Mode	TSTRN	130
H029	Check Run Mode	CHKRN	131
H030	Feedhold	FHLD1	132
H031	Cycle In Progress	CYCIP	133
H032	Postlude (Post Move)	PSTMV	134
H033	M Strobe	MSTR	135
H034	S Strobe 1	SSTR1	136
H035	NOT USED	-----	137

G Variable 001-040

G variables carry values (16 bit word) from the CNC executive to PAL. In PAL G variables are Read only. The PAL ladder cannot write (reply) to the CNC using a G variable.

The following list is an example of the information that the G variable carries to PAL.

Important: This is not a comprehensive list of G variables, it is only an example of the types used in the 8400 PAL. Check your PAL flags and variables manual to find the G variables for your application flags and variables.

--- LATHE INDEX ---

G FLAGS - READ ONLY

ADDRESS	NAME	MNEMONIC	MEMORY PAGE
G001	Programmed M Value	MFUNC	180
G002	Programmed S Word Value 1	SPSP1	181
G003	NOT USED	-----	182
G004	Programmed T Offset Value 1	TLND1	183
G005	Programmed T Word Value 2	TLND2	184
G006	Current CNC Mode	MODE	185
G007	Current Spindle Speed	S1RPM	186

G008 THROUGH G040 NOT ALLOCATED

N Variable 001-100

The N flag is a bit flag (on or off) to the CNC executive that is Write/Read. The PAL can look at, set and clear this flag; the CNC executive can only look at it. The Pal ladder can look at this flag and it can write (reply) to the CNC using this flag. The N flag may be used with the H flag.

The N flag's main objective is to request the CNC to perform an activity for the PAL or to advise the CNC of certain conditions.

The following list is an example of the information that the N flag carries to the CNC executive describing a specific PAL condition.

Important: This is not a comprehensive list of N flags. It is only an example of the types of flags used in the 8400 PAL. Check your PAL flags and variables manual to find the N flags for your application.

--- LATHE INDEX ---

N FLAGS - READ/WRITE

ADDRESS	NAME	MNEMONIC	MEMORY PAGE
N041	NOT USED	-----	042
N042	T Reply 1	TRPY1	043
N043	T Reply 2	TRPY2	044
N044	Axis 1 Permit Motion	AX1CO	045
N045	Axis 2 Permit Motion	AX2CO	046
N046	Axis 3 Permit Motion	AX3CO	047
N047	Axis 4 Permit Motion	AX4CO	048
N048	Axis 5 Permit Motion	AX5CO	049
N049	Axis 6 Permit Motion	AX6CO	050
N050	Axis 7 Permit Motion	AX7CO	051
N051	Axis 8 Permit Motion	AX8CO	052
N052	Inhibit Constant Surface Speed	IHSS	053
N053	Spindle Clockwise Request (M03)	S1FWD	054
N054	Spindle Counterclockwise Request (M04)	S1RVS	055
N055	NOT USED	-----	056
N056	NOT USED	-----	057
N057	Analog Spindle Control	ASC	058
N058	Cycle Start Inhibit	CAUTION	059
N059	Keyboard Lockout Request	KBDLK	060
N060	No Axis Wait For M Reply	FLY-M	061

L Variable 001-040

The L variable is a value flag (16 bit word, not an on/off bit) sent to the CNC executive from PAL. The PAL ladder can look at, set and clear this flag; the CNC executive can only look at it. L is a Write/Read variable PAL ladder can look at and write to the CNC with. The L flag may be used with the G flag.

The L variables main objective is to provide numerical information for certain CNC functions.

The following list is an example of the information that the L variable carries to the PAL ladder describing the mode of operation.

Important: This is not a comprehensive list of L variables. It is only an example of the types of variables used in the 8400 PAL. Check your PAL flags and variables manual to find the L flags for your application.

--- LATHE INDEX ---

L WORDS - READ/WRITE

ADDRESS	NAME	MNEMONIC	MEMORY PAGE
L001	Feedrate Override Value	FEERO	089
L002	Spindle Speed Demand Value	S1REQ	090
L003	Spindle Scaling Value (RPM/10V)	S1GRT	091
L004	Spindle Speed Override Value	SPOVR	092
L005 THROUGH L036 ARE NOT ALLOCATED		-----	093
L037	Arithmetic Value for Message M037	L037	099
L038	Arithmetic Value for Message M038	L038	100
L039	Arithmetic Value for Message M039	L039	101
L040	Arithmetic Value for Message M040	L040	102

M Flags
001-040

The M flag is a bit flag to the CNC executive that is Write/Read. The PAL ladder can request this flag and it can write to the CNC using this flag. The M flag turns on stored messages in PAL that will display on the CNC control CRT screen.

There are three types of error messages:

- emergency
- Status
- variable

A catastrophic fault such as a motor overspeed causes PAL to display emergency messages on the CNC. The fault causes a message to be displayed and the machine to go into E-Stop.

Conditions causing a feedhold condition such as axis over travel causes PAL to display status messages on the CNC. The fault causes a message to be displayed and the machine to go into feedhold.

These messages are not generated by M flags, hardware overtravels, etc., though detected by PAL and sent to the control. N flags use messages buried in the exec. The only PAL flag that causes an E-Stop is the E-Stop N flag. In PAL there are "ALARM" (see page 4-11), STATUS, and VARIABLE messages. Read section 4.4 of this manual.

PAL displays variable messages when requested by the task program or PAL. The request causes the message to be displayed anytime. These messages are to prompt the machine operator to perform specific functions or to inform him of a condition.

Timers
T001-T0016

Timer instructions are used in PAL to delay the start time of an activity or series of activities. Timer instructions:

- are always part of an output group in a rung
- are edge triggered by the false to true change in preconditioning a rung
- delay an activity until a required condition has been achieved
- are only used with examine on or examine off instructions

There are 16 timers that you can use in PAL. The timer instructions address these timers.

Timers are turned on or off according to the preconditions of the rung they are in. The preconditions must go from false to true for a timer instruction to have an effect. You must control the preconditions of a rung that controls a timer to make the timer do what you want.

Counters
C001-C016

Counter instructions are used in PAL to repeat an activity or group of activities a specific number of times. Counter instructions:

- are always part of an output group in a rung
- are edge triggered by the false to true change when preconditioning a rung
- address the counters

There are 16 counters that you can use in PAL. Each counter can count up to 65335.

Counters count up or down according to the preconditions of the rung they are in. The preconditions must go from false to true for a counter instruction to have any effect. You must control the preconditions of a rung that controls a counter to make the counters do what you want.

Inside PAL when more than simple bit logic is needed, the software uses timers, counters, storage areas, and arithmetic functions during the execution of the PAL ladder diagram.

Figure 2-1 shows how the CNC executive, the PAL software, the PAL ladder diagram and the discrete I/O work together using the flags addresses, timers and counters.

2.2 Chapter Summary

This chapter introduced you to the PAL ladder diagram and how the PAL commuicates with the system. Flags and Variables used in PAL were defined and illustrated using examples. The difference between Flags and Variables and Timers and Counters will help you when writing the PAL ladder examples in the next chapters.

3.0 Programming Guidelines and Examples

This section offers a few suggestions for making your 8400 PAL development easier. If you understand what is given here, and adapt it to your own way of developing 8400 PAL, you should be a more effective PAL programmer.

To use this section, you should:

- Know what 8400 PAL is (see section 1).
- Be familiar with the equipment for 8400 PAL development (see section 2).
- Be familiar with how to operate the Kaypro with CP/M (this last requirement you'll have to learn outside this manual).

Writing a PAL program is not difficult if you properly prepare your data beforehand. Clearly defined machining objectives, a working knowledge of your Series 8400 control, and the instructions in this manual will prepare you for writing PAL programs.

This section is organized according to the steps you take in developing 8400 PAL. If this is your first time through the process, here is the order we recommend:

We recommend that you understand this section before moving on to actual programming.

3.1 Organizing the Development System

The first recommendation we can make is to organize your development system.

You want to be efficient when you program 8400 PAL. Establish a way of doing things that lets you go through the process (and correct any errors) quickly.

To get organized and be efficient, you must know how to operate the Kaypro. You must be able to maintain, move, and manipulate files and programs using the CP/M disk operating system. Take the time to learn as much as possible about the computer (it will save you time later on). It is a useful tool for more than just 8400 PAL programming.

Organize Your Development Disks

The standard PAL development software (cat. no. 8400-PAL) comes on a 5 1/4 inch floppy disk. This disk contains:

- The PAL program (revision Y28) for writing PAL programs and converting the ladder source code - what you write on the Kaypro, to object code - and what the CNC executes. It also allows you to download object code to the 8400 for testing, transfer object code to the prom programmer for EPROM burning, and print out your ladder diagram.

- The XREFLAD.BAS program. This program is given on the disk in source code. You must compile this program using S-Basic to get a XREFLAD.COM program. This program will allow you to generate and print out a cross reference of flags, variables, rungs and instructions in the ladder program.

Make a backup copy of this disk.

If you are going to be writing a lot of PAL programs, we suggest you make and use a two disk system. The two disk system consists of:

- a Main Disk that contains the CP/M operating system, STAT.COM, XREFLAD.COM, and the PAL.COM program

- a Data Disk that contains PAL source code files

Make backup copies of these disks.

The Main Disk can be write protected, but not the other disk (except for backup copies, of course) since it is necessary to write to the Data Disk during PAL development.

During development, the Main Disk always goes in drive A, and the Data Disk always goes in drive B. This procedure is recommended because it's compatible with the steps that we give you later on in this manual.

Organize Your Development Hardware

This step is simply a matter of obtaining the hardware you need for the process, and making sure it works properly. In general, you need to have:

- an 8400 CNC that has a CRT and keyboard, connected to the system power supply
- input and output headers for the sockets on the boards in the Control Module (I/O DIPS 8400-Z1 and 8400-Z0) This is needed for the Series A Hardware only, not for the Series B Hardware.
- cables from the Control Module to the input/output boards (8400-CX5, or 8400CX10)
- input/output boards for the application (8400-XS1 or 8400-XQD)
- isolated input and output modules for the application (8400-XSA1, XSA0, XSD1, XSD0, XQA1, XQA0, XQD1, XQD0)
- the Kaypro and your development disks (8400-XK and 8400-PALE or 8400-XPG)
- the download cable from the Kaypro to the 8400 (8400-XPC3 or the 8400-XPC5 which is used with the peripheral panel)
- the cable from the Kaypro to the prom programmer (8400-XPC2)
- the prom programmer (8400-XPB)
- a supply of EPROMS (2716's for 8400 R, GP, MP, Bandit III & BANDIT IV; 2764's for 8400 GLC and Rev B hardware for BIV). (8000-X16)

We also recommend that you make, and use, a test fixture to test your PAL programs before they are used on an active machine. The design of this test fixture is completely up to you, but it should have the isolated inputs and outputs, switches and indicator lights necessary to test your application of 8400 PAL.

3.2 Allocating Inputs and Outputs

The first step in developing 8400 PAL is to study your application and do some thorough planning. You should know:

- the functional definitions and requirements for the external devices to be controlled (spindle, turret, coolant, etc.). This should include timing diagrams, if required
- the requirements and restrictions for M, S, and T codes as indicated by functional definitions
- the requirements for feedrate and spindle speed override, Emergency Stop, axis homing and overtravel, and miscellaneous operator switches and buttons

With your application understood, you can proceed with assigning inputs and outputs in the system. Here are two steps and things you should know about to plan system inputs and outputs:

1. Make a list of all inputs and outputs.

   Make a list of all inputs and outputs to be used by PAL. Decide which type of input/output board you are going to use (the example that follows assumes that you will use an 8000-XQBD or 8400-XS1). Decide which PAL ports will be inputs and which will be outputs.

   On the Series A 8400, there are 5 connectors in the Control Module that can be used by PAL. In the revision A hardware Control Module, there are 3 connectors on the CRT module, and 2 on the CPU Module that are allocated for PAL. In the revision B hardware Control Module, there are 5 connectors on the CRT Module only, and no connectors on the CPU Module that are allocated for PAL.

   You should use the connectors on the CRT module first when allocating inputs and outputs.

   Each connector has 24 I/O points (50 pins). There are 3 PAL ports for each connector, so each PAL port is made up of 8 I/O points. Each I/O point has a corresponding R flag assigned in PAL. There are 120 R flags.

   When you allocate I/O for PAL, you decide which PAL ports are inputs and which are outputs. You decide, in effect, which groups of 8 R flags are inputs, and which groups of 8 are outputs.

Decide which PAL ports are inputs and which are outputs, then, only if you have revision A hardware, install headers in IC locations for each PAL port that you will use. There are two locations for IC headers for each PAL port.

If the port is an input port, you must install input IC's (BLC902's -- 4 inputs, resistor networks, catalog number 8400-Z1).

If the port is an output, you must install output IC's (7409's -- 4 outputs, quad AND gates catalog number 8400-Z0).

See the tables in step 2 to organize your I/O allocations.

Important: For series A Hardware, PAL ports 01 through 06 are located on the CPU Module. This gives a total of 120 I/O points.

2. Make a diagram of your application and I/O assignments.

   Make a diagram of the I/O boards showing each station, function, input/output module, and external connection. For example, see figure 3-1.

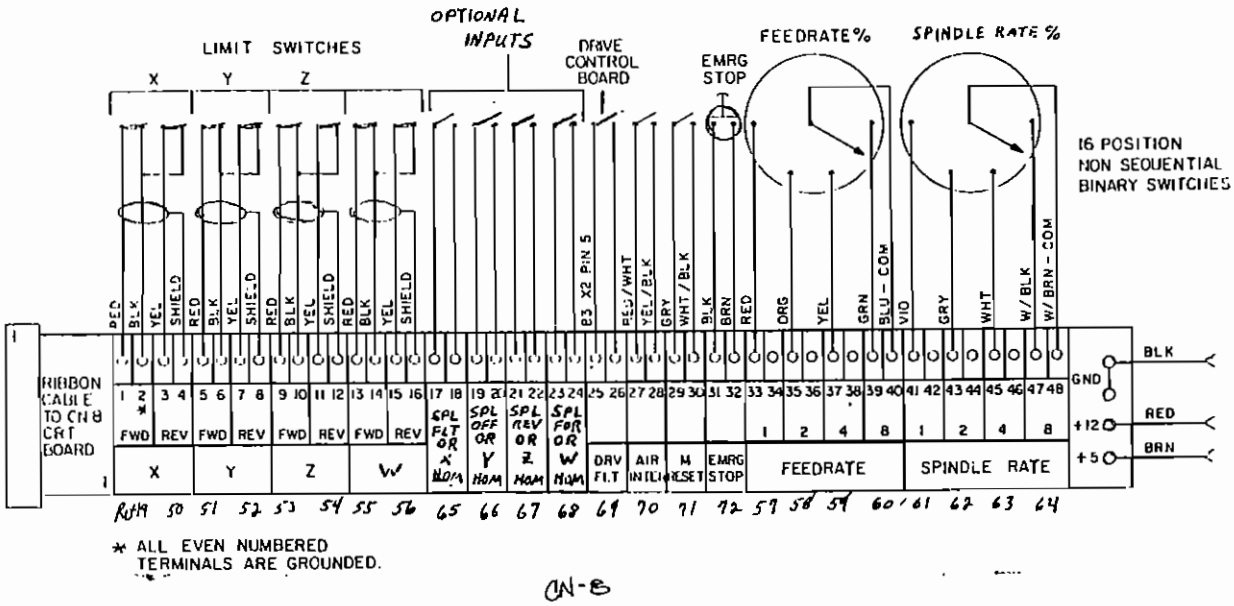


Figure 3-1: I/O Board Diagram

You should now have a very good idea of how your I/O allocations will go. Use the following tables on pages 3-7 through 3-11 for B Hardware and pages 3-12 through 3-16 for A Hardware to summarize your I/O allocations.

One thing that we haven't mentioned yet about I/O allocations is that any two PAL ports can be used in a V variable. This puts 16 R flags into each V variable. When you assign V variables for PAL, you specify which port will make up the most significant byte of the V variable, and which will be the least significant byte. Even though you assign 16 R flags to each V variable, they can still be used as individual R flags. You don't have to use all 16 bits of the V variable.

Series B Hardware 8400 I/O Assignments

PAL Ports 01, 02, and 03 - CRT-P3

CRT Board, Connector P3

PAL Port Numbers (Input or Output)	PAL R Flags	CPU Pin Out	Opto 22 I/O Cable Pins	Opto 22 I/O Board Indicators	Opto 22 Module Terminals	Description of I/O Device or Connection
	R016	PB7	47	0 (_____)	1 (_____)	
	R015	PB6	45	1 (_____)	3 (_____)	
	R014	PB5	43	2 (_____)	5 (_____)	
02	R013	PB4	41	3 (_____)	7 (_____)	
(_____)	R012	PB3	39	4 (_____)	9 (_____)	
	R011	PB2	37	5 (_____)	11 (_____)	
	R010	PB1	35	6 (_____)	13 (_____)	
	R009	PB0	33	7 (_____)	15 (_____)	
	R024	PC7	31	8 (_____)	17 (_____)	
	R023	PC6	29	9 (_____)	19 (_____)	
	R022	PC5	27	10 (_____)	21 (_____)	
03	R021	PC4	25	11 (_____)	23 (_____)	
(_____)	R020	PC3	23	12 (_____)	25 (_____)	
	R019	PC2	21	13 (_____)	27 (_____)	
	R018	PC1	19	14 (_____)	29 (_____)	
	R017	PC0	17	15 (_____)	31 (_____)	
	R008	PA7	15	16 (_____)	33 (_____)	
	R007	PA6	13	17 (_____)	35 (_____)	
	R006	PA5	11	18 (_____)	37 (_____)	
01	R005	PA4	9	19 (_____)	39 (_____)	
(_____)	R004	PA3	7	20 (_____)	41 (_____)	
	R003	PA2	5	21 (_____)	43 (_____)	
	R002	PA1	3	22 (_____)	45 (_____)	
	R001	PA0	1	23 (_____)	47 (_____)	

Solid State Relay types:	Quad Packs	Single
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series B Hardware 8400 I/O Assignments

PAL Ports 04, 05, and 06 - CRT-P4

CRT Board, Connector P4

PAL Port Numbers (Input or Output)	PAL R Flags	CPU Pin Out Out	Opto 22 I/O Cable Pins	Opto 22 I/O Board Indicators	Opto 22 Module Terminals	Description of I/O Device or Connection
	R040	PB7	47	0 (_____)	1 (_____)	
	R039	PB6	45	1 (_____)	3 (_____)	
	R038	PB5	43	2 (_____)	5 (_____)	
05	R037	PB4	41	3 (_____)	7 (_____)	
(_____)	R036	PB3	39	4 (_____)	9 (_____)	
	R035	PB2	37	5 (_____)	11 (_____)	
	R034	PB1	35	6 (_____)	13 (_____)	
	R033	PB0	33	7 (_____)	15 (_____)	
	R048	PC7	31	8 (_____)	17 (_____)	
	R047	PC6	29	9 (_____)	19 (_____)	
	R046	PC5	27	10 (_____)	21 (_____)	
06	R045	PC4	25	11 (_____)	23 (_____)	
(_____)	R044	PC3	23	12 (_____)	25 (_____)	
	R043	PC2	21	13 (_____)	27 (_____)	
	R042	PC1	19	14 (_____)	29 (_____)	
	R041	PC0	17	15 (_____)	31 (_____)	
	R032	PA7	15	16 (_____)	33 (_____)	
	R031	PA6	13	17 (_____)	35 (_____)	
	R030	PA5	11	18 (_____)	37 (_____)	
04	R029	PA4	9	19 (_____)	39 (_____)	
(_____)	R028	PA3	7	20 (_____)	41 (_____)	
	R027	PA2	5	21 (_____)	43 (_____)	
	R026	PA1	3	22 (_____)	45 (_____)	
	R025	PA0	1	23 (_____)	47 (_____)	

Solid State Relay types:	Quad Packs	Single
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series B Hardware 8400 I/O Assignments

PAL Ports 07, 08, and 09 - CRT-P7

CRT Board, Connector P7

PAL Port Numbers (Input or Output)	PAL R Flags	CPU Pin Out	Opto 22 I/O Cable Pins	Opto 22 I/O Board Indicators	Opto 22 Module Terminals	Description of I/O Device or Connection
	R064	PB7	47	0 (_____)	1 (_____)	
	R063	PB6	45	1 (_____)	3 (_____)	
	R062	PB5	43	2 (_____)	5 (_____)	
08	R061	PB4	41	3 (_____)	7 (_____)	
(_____)	R060	PB3	39	4 (_____)	9 (_____)	
	R059	PB2	37	5 (_____)	11 (_____)	
	R058	PB1	35	6 (_____)	13 (_____)	
	R057	PB0	33	7 (_____)	15 (_____)	
	R072	PC7	31	8 (_____)	17 (_____)	
	R071	PC6	29	9 (_____)	19 (_____)	
	R070	PC5	27	10 (_____)	21 (_____)	
09	R069	PC4	25	11 (_____)	23 (_____)	
(_____)	R068	PC3	23	12 (_____)	25 (_____)	
	R067	PC2	21	13 (_____)	27 (_____)	
	R066	PC1	19	14 (_____)	29 (_____)	
	R065	PC0	17	15 (_____)	31 (_____)	
	R056	PA7	15	16 (_____)	33 (_____)	
	R055	PA6	13	17 (_____)	35 (_____)	
	R054	PA5	11	18 (_____)	37 (_____)	
07	R053	PA4	9	19 (_____)	39 (_____)	
(_____)	R052	PA3	7	20 (_____)	41 (_____)	
	R051	PA2	5	21 (_____)	43 (_____)	
	R050	PA1	3	22 (_____)	45 (_____)	
	R049	PA0	1	23 (_____)	47 (_____)	

Solid State Relay types:	Quad Packs	Single
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series B Hardware 8400 I/O Assignments

PAL Ports 10, 11, and 12 - CRT-P6

CRT Board, Connector P6

PAL Port Numbers (Input or Output)	PAL R Flags	CPU Pin Out	Opto 22 I/O Cable Pins	Opto 22 I/O Board Indicators	Opto 22 Module Terminals	Description of I/O Device or Connection
	R088	PB7	47	0 (_____)	1 (_____)	
	R087	PB6	45	1 (_____)	3 (_____)	
	R086	PB5	43	2 (_____)	5 (_____)	
11	R085	PB4	41	3 (_____)	7 (_____)	
(_____)	R084	PB3	39	4 (_____)	9 (_____)	
	R083	PB2	37	5 (_____)	11 (_____)	
	R082	PB1	35	6 (_____)	13 (_____)	
	R081	PB0	33	7 (_____)	15 (_____)	
	R096	PC7	31	8 (_____)	17 (_____)	
	R095	PC6	29	9 (_____)	19 (_____)	
	R094	PC5	27	10 (_____)	21 (_____)	
12	R093	PC4	25	11 (_____)	23 (_____)	
(_____)	R092	PC3	23	12 (_____)	25 (_____)	
	R091	PC2	21	13 (_____)	27 (_____)	
	R090	PC1	19	14 (_____)	29 (_____)	
	R089	PC0	17	15 (_____)	31 (_____)	
	R080	PA7	15	16 (_____)	33 (_____)	
	R079	PA6	13	17 (_____)	35 (_____)	
	R078	PA5	11	18 (_____)	37 (_____)	
10	R077	PA4	9	19 (_____)	39 (_____)	
(_____)	R076	PA3	7	20 (_____)	41 (_____)	
	R075	PA2	5	21 (_____)	43 (_____)	
	R074	PA1	3	22 (_____)	45 (_____)	
	R073	PA0	1	23 (_____)	47 (_____)	

Solid State Relay types:	Quad Packs	Single
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series B Hardware 8400 I/O Assignments

PAL Ports 13, 14, and 15 - CRT-P5

CRT Board, Connector P5

PAL Port Numbers (Input or Output)	PAL R Flags	CPU Pin Out	Opto 22 I/O Cable Pins	Opto 22 I/O Board Indicators	Opto 22 Module Terminals	Description of I/O Device or Connection
	R112	PB7	47	0 (_____)	1 (_____)	
	R111	PB6	45	1 (_____)	3 (_____)	
	R110	PB5	43	2 (_____)	5 (_____)	
14	R109	PB4	41	3 (_____)	7 (_____)	
(_____)	R108	PB3	39	4 (_____)	9 (_____)	
	R107	PB2	37	5 (_____)	11 (_____)	
	R106	PB1	35	6 (_____)	13 (_____)	
	R105	PB0	33	7 (_____)	15 (_____)	
	R120	PC7	31	8 (_____)	17 (_____)	
	R119	PC6	29	9 (_____)	19 (_____)	
	R118	PC7	27	10 (_____)	21 (_____)	
15	R117	PC4	25	11 (_____)	23 (_____)	
(_____)	R116	PC3	23	12 (_____)	25 (_____)	
	R115	PC2	21	13 (_____)	27 (_____)	
	R114	PC1	19	14 (_____)	29 (_____)	
	R113	PC0	17	15 (_____)	31 (_____)	
	R104	PA7	15	16 (_____)	33 (_____)	
	R103	PA6	13	17 (_____)	35 (_____)	
	R102	PA5	11	18 (_____)	37 (_____)	
13	R101	PA4	9	19 (_____)	39 (_____)	
(_____)	R100	PA3	7	20 (_____)	41 (_____)	
	R099	PA2	5	21 (_____)	43 (_____)	
	R098	PA1	3	22 (_____)	45 (_____)	
	R097	PA0	1	23 (_____)	47 (_____)	

Solid State Relay types:	Quad Packs	Single
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series A Hardware 8400 I/O Assignments - PAL Ports 07, 08, and 09 - CRT-CN8

CRT Board, Connector CN8

PAL Port Numbers (Input or Output)	IC Numbers (IC Type)	PAL R-Flags	I/O Cable Pins	Opto 22 Board I/O Indicators / I/O Terminals (Opto 22 Module)	(Description of I/O Connection or Device)
		R064	47	0 (_____) 1	(_____)
	IC 159	R063	45	1 (_____) 3	(_____)
	(_____)	R062	43	2 (_____) 5	(_____)
08		R061	41	3 (_____) 7	(_____)
(_____)		R060	39	4 (_____) 9	(_____)
	IC 161	R059	37	5 (_____) 11	(_____)
	(_____)	R058	35	6 (_____) 13	(_____)
		R057	33	7 (_____) 15	(_____)
		R072	31	8 (_____) 17	(_____)
	IC 125	R071	29	9 (_____) 19	(_____)
	(_____)	R070	27	10 (_____) 21	(_____)
09		R069	25	11 (_____) 23	(_____)
(_____)		R068	23	12 (_____) 25	(_____)
	IC 122	R067	21	13 (_____) 27	(_____)
	(_____)	R066	19	14 (_____) 29	(_____)
		R065	17	15 (_____) 31	(_____)
		R056	15	16 (_____) 33	(_____)
	IC 124	R055	13	17 (_____) 35	(_____)
	(_____)	R054	11	18 (_____) 37	(_____)
07		R053	9	19 (_____) 39	(_____)
(_____)		R052	7	20 (_____) 41	(_____)
	IC 120	R051	5	21 (_____) 43	(_____)
	(_____)	R050	3	22 (_____) 45	(_____)
		R049	1	23 (_____) 47	(_____)

Solid State Relay types:	Quad Packs Cat. No.	Single Cat. No.	IC Types:	Cat. No.
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1	4 Inputs	8400-Z1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0	4 Outputs	8400-Z0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1		
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0		

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. If you have two ICs in a port, they must be identical.

Series A Hardware 8400I/O Assignments - PAL Ports 13, 14, and 15 - CRT-CN5

CRT Board, Connector CN5

PAL Port Numbers (Input or Output)	IC Numbers (IC Type)	PAL R-Flags		Opto 22 Board I/O Indicators	I/O Terminals (Opto 22 Module)		(Description of I/O Connection or Device)
		R112	47	0	(_____)	1	(_____)
	IC 87	R111	45	1	(_____)	3	(_____)
	(_____)	R110	43	2	(_____)	5	(_____)
14		R109	41	3	(_____)	7	(_____)
(_____)		R108	39	4	(_____)	9	(_____)
	IC 90	R107	37	5	(_____)	11	(_____)
	(_____)	R106	35	6	(_____)	13	(_____)
		R105	33	7	(_____)	15	(_____)
		R120	31	8	(_____)	17	(_____)
	IC 89	R119	29	9	(_____)	19	(_____)
	(_____)	R118	27	10	(_____)	21	(_____)
15		R117	25	11	(_____)	23	(_____)
(_____)		R116	23	12	(_____)	25	(_____)
	IC 86	R115	21	13	(_____)	27	(_____)
	(_____)	R114	19	14	(_____)	29	(_____)
		R113	17	15	(_____)	31	(_____)
		R104	15	16	(_____)	33	(_____)
	IC 88	R103	13	17	(_____)	35	(_____)
	(_____)	R102	11	18	(_____)	37	(_____)
13		R101	9	19	(_____)	39	(_____)
(_____)		R100	7	20	(_____)	41	(_____)
	IC 52	R099	5	21	(_____)	43	(_____)
	(_____)	R098	3	22	(_____)	45	(_____)
		R097	1	23	(_____)	47	(_____)

I/O Cable Pins

Solid State Relay types:	Quad Packs Cat. No.	Single Cat. No.	IC Types:	Cat. No.
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1	4 Inputs	8400-Z1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0	4 Outputs	8400-Z0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1		
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0		

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. Therefore, if you have two ICs in a port, they must be identical.

Series A Hardware 8400 I/O Assignments - PAL Ports 11, 12, 10 - CRT-CN10

CRT Board, Connector CN10

PAL Port Numbers (Input or Output)	IC Numbers (IC Type)	PAL R-Flags		Opto 22 Board I/O Indicators ¦ I/O Terminals (Opto 22 Module)		(Description of I/O Connection or Device)
		R088	47	0 (_____)	1	(_____)
	IC 201	R087	45	1 (_____)	3	(_____)
	(_____)	R086	43	2 (_____)	5	(_____)
11		R085	41	3 (_____)	7	(_____)
(_____)		R084	39	4 (_____)	9	(_____)
	IC 205	R083	37	5 (_____)	11	(_____)
	(_____)	R082	35	6 (_____)	13	(_____)
		R081	33	7 (_____)	15	(_____)
		R096	31	8 (_____)	17	(_____)
	IC 204	R095	29	9 (_____)	19	(_____)
	(_____)	R094	27	10 (_____)	21	(_____)
12		R093	25	11 (_____)	23	(_____)
(_____)		R092	23	12 (_____)	25	(_____)
	IC 200	R091	21	13 (_____)	27	(_____)
	(_____)	R090	19	14 (_____)	29	(_____)
		R089	17	15 (_____)	31	(_____)
		R080	15	16 (_____)	33	(_____)
	IC 203	R079	13	17 (_____)	35	(_____)
	(_____)	R078	11	18 (_____)	37	(_____)
10		R077	9	19 (_____)	39	(_____)
(_____)		R076	7	20 (_____)	41	(_____)
	IC 160	R075	5	21 (_____)	43	(_____)
	(_____)	R074	3	22 (_____)	45	(_____)
		R073	1	23 (_____)	47	(_____)

I/O Cable Pins ¦

Solid State Relay types:	Quad Packs Cat. No.	Single Cat. No.	IC Types:	Cat. No.
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1	4 Inputs	8400-Z1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0	4 Outputs	8400-Z0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1		
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0		

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. Therefore, if you have two ICs in a port, they must be identical.

Series A Hardware 8400 I/O Assignments - PAL Ports 01, 02, 03 - CPU-CN15

CPU Board, Connector CN15

PAL Port Numbers (Input or Output)	IC Numbers (IC Type)	PAL R-Flags	I/O Terminals	Opto 22 Board I/O Indicators (Opto 22 Module)	(Description of I/O Connection or Device)
		R016	47 0	( ) 1	( )
	IC 157	R015	45 1	( ) 3	( )
	( )	R014	43 2	( ) 5	( )
02		R013	41 3	( ) 7	( )
( )		R012	39 4	( ) 9	( )
	IC 156	R011	37 5	( ) 11	( )
	( )	R010	35 6	( ) 13	( )
		R009	33 7	( ) 15	( )
	IC 160	R024	31 8	( ) 17	( )
	IC 155	R023	29 9	( ) 19	( )
	IC 160	R022	27 10	( ) 21	( )
03	IC 155	R021	25 11	( ) 23	( )
( )	IC 155	R020	23 12	( ) 25	( )
	( )	R019	21 13	( ) 27	( )
	IC 160	R018	19 14	( ) 29	( )
	( )	R017	17 15	( ) 31	( )
		R008	15 16	( ) 33	( )
	IC 159	R007	13 17	( ) 35	( )
	( )	R006	11 18	( ) 37	( )
01		R005	9 19	( ) 39	( )
( )		R004	7 20	( ) 41	( )
	IC 158	R003	5 21	( ) 43	( )
	( )	R002	3 22	( ) 45	( )
		R001	1 23	( ) 47	( )

I/O Cable Pins

Solid State Relay types:	Quad Packs Cat. No.	Single Cat. No.	IC Types:	Cat. No.
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1	4 Inputs	8400-Z1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0	4 Outputs	8400-Z0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1		
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0		

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. Therefore, if you have two ICs in a port, they must be identical.

Series A Hardware 8400I/O Assignments - PAL Ports 04, 05, 06 - CPU-CN16

CPU Board, Connector CN16

PAL Port Numbers (Input or Output)	IC Numbers (IC Type)	PAL R-Flags		Opto 22 Board I/O Indicators	I/O Terminals (Opto 22 Module)		(Description of I/O Connection or Device)
		R040	47	0	(_____)	1	(_____)
	IC 166	R039	45	1	(_____)	3	(_____)
	(_____)	R038	43	2	(_____)	5	(_____)
05		R037	41	3	(_____)	7	(_____)
(_____)		R036	39	4	(_____)	9	(_____)
	IC 165	R035	37	5	(_____)	11	(_____)
	(_____)	R034	35	6	(_____)	13	(_____)
		R033	33	7	(_____)	15	(_____)
	IC 169	R048	31	8	(_____)	17	(_____)
	IC 164	R047	29	9	(_____)	19	(_____)
	IC 169	R046	27	10	(_____)	21	(_____)
06	IC 164	R045	25	11	(_____)	23	(_____)
(_____)	IC 164	R044	23	12	(_____)	25	(_____)
	(_____)	R043	21	13	(_____)	27	(_____)
	IC 169	R042	19	14	(_____)	29	(_____)
	(_____)	R041	17	15	(_____)	31	(_____)
		R032	15	16	(_____)	33	(_____)
	IC 168	R031	13	17	(_____)	35	(_____)
	(_____)	R030	11	18	(_____)	37	(_____)
04		R029	9	19	(_____)	39	(_____)
(_____)		R028	7	20	(_____)	41	(_____)
	IC 167	R027	5	21	(_____)	43	(_____)
	(_____)	R026	3	22	(_____)	45	(_____)
		R025	1	23	(_____)	47	(_____)

I/O Cable Pins ¦

Solid State Relay types:	Quad Packs Cat. No.	Single Cat. No.	IC Types:	Cat. No.
IDC5 = 24V DC Input	8000-XQD1	8400-XSD1	4 Inputs	8400-Z1
ODC5 = 24V DC Output	8000-XQD0	8400-XSD0	4 Outputs	8400-Z0
IAC5 = 120V AC Input	8000-XQA1	8400-XSA1		
OAC5 = 120V AC Output	8000-XQA0	8400-XSA0		

You may use one or two ICs for each port. A port can only have four or eight inputs, or four or eight outputs. It cannot have four inputs and four outputs. Therefore, if you have two ICs in a port, they must be identical.

3.3 Preprogramming Flowcharting

Once you have planned and allocated the inputs and outputs for PAL, you can proceed with planning your PAL program. This is what we call "preprogramming." There are two aspects to preprogramming: flowcharting and sketching the ladder.

The first thing you should do in preprogramming is to make a flowchart of your PAL program. It gives you a base for a systematic approach to PAL programming, it gives structure to your PAL programs and allows you to add steps easily when they are required.

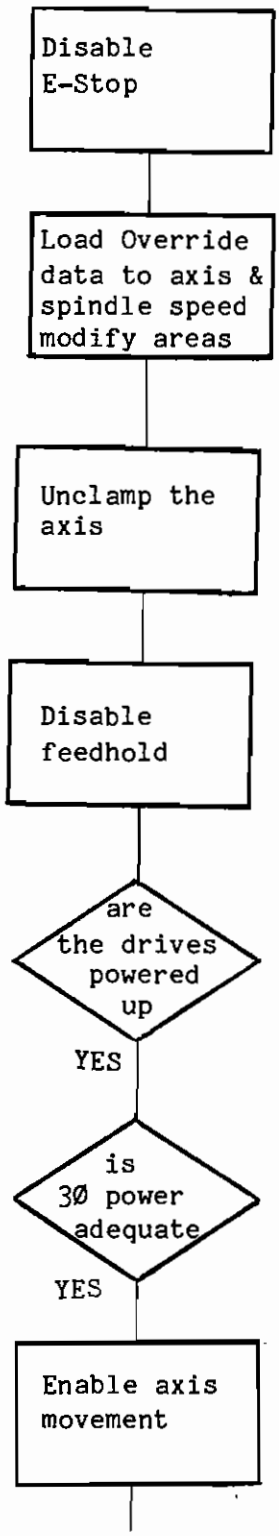
The following flowchart is an example of a start up ladder for the B series 8400 GP control. This is part of a ladder available for the 8400 Control. Appendix D is a more comprehensive example of PAL ladder logic. It contains several conditions not found in a simple start up ladder.

Study these flowcharts to get an idea of how 8400 PAL programs are structured.

One thing that affects 8400 PAL structure is the time spent executing PAL and the flow of execution. Remember that 8400 PAL takes 2 to 6 ms. to execute, and the ladder is executed every 40 ms. You can't have a program that takes a lot of time.

Important: You cannot jump "backward" in PAL, and each execution of the ladder must flow through from beginning to end. Your program must copy the state of inputs, make required tests, do required actions, and be ready for the next pass. If an action is not complete on one pass through PAL (PAL may be waiting for a switch closure, a certain count, or a timed out timer), you must set flags to tell PAL where to begin again on the next pass.

Flowchart for Series B 8400 GP Control start up PAL



This example ends here

Sketching the Ladder

After you have a good flowchart for your ladder, you should sketch the ladder program.

Sketch the ladder the way you will enter it using the Kaypro.

Outline your use of R, B, N, and H flags, and assign mnemonics to each one. Also specify your use of V, L, and G variables and assign mnemonics to these as well. If you use counters or timers, assign mnemonics to these. Use the mnemonics in the sketch.

Write out your assignment of messages, M flags, their text and associated L variables (used with variable messages).

The following list contains several standard functions you will probably use in any 8400 PAL program.

- Fault Clear to PAL from the CNC
- Emergency Stop switch monitoring
- Feedrate Override switch monitoring
- Drives On
- Drives OK
- Axis Move Enable (clamps off)
- Home switch monitoring
- Overtravel switch monitoring
- M-functions (pre- and post-move)

M2 - End of Program
M3 - Spindle On Forward
M4 - Spindle On Reverse
M5 - Spindle Off
M8 - Coolant On
M9 - Coolant Off
M12 - Delay M-function
M21 - Wait for Reset M-function
    - Spindle on/off control
(no speed control in this example)

Tool functions are not implemented in this example.

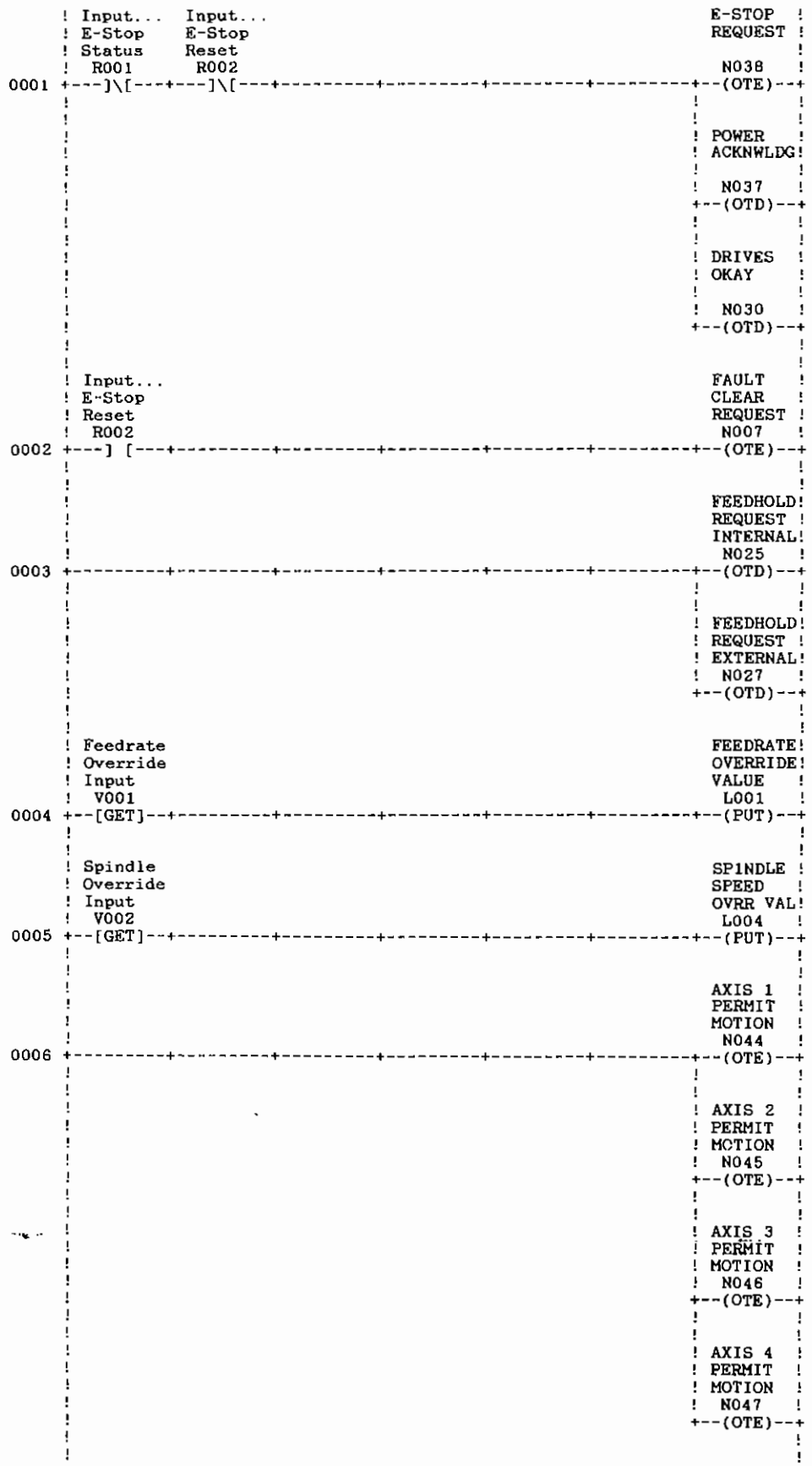
The following two examples are the minimum start up ladders for the 8400 Series CLC and GP controls. These programs are intended for your information only. In no way does this represent a fully functional ladder.

We recommend that you use these ladders and the ladder in Appendix F to practice using the Kaypro, and to study the concepts involved in PAL development.

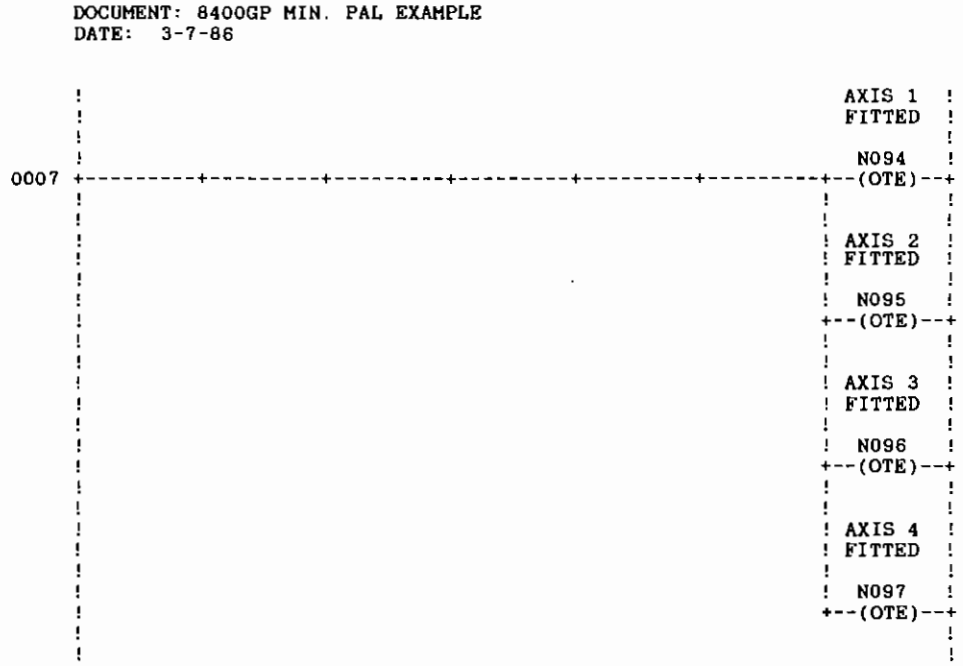
The list provided with the example in Appendix F was generated using a development system that is more sophisticated than the one provided with the Kaypro. This is why you may notice differences between what is given, and what you write on the Kaypro.

```
DOCUMENT: 8400GLC MIN. PAL EXAMPLE
DATE:  3-7-86

     ! Input...  Input...                                          E-STOP  !
     ! E-Stop    Frnt Pnl                                        REQUEST !
     ! Status    FLT/CLR                                                 !
     !  R001      R002                                            N038   !
0001 +---]\[---+---]\[---+---------+---------+---------+---------+--(OTE)--+
     !                                                                    !
     !                                                                    !
     !                                                          ! POWER   !
     !                                                          ! ACKNWLDG!
     !                                                          !         !
     !                                                          !  N037   !
     !                                                          +--(OTD)--+
     !                                                          !         !
     !                                                          !         !
     !                                                          ! DRIVES  !
     !                                                          ! OKAY    !
     !                                                          !         !
     !                                                          !  N030   !
     !                                                          +--(OTD)--+
     !                                                                    !
     !                                                                    !
     ! Feedrate                                                 FEEDRATE!
     ! Override                                                 OVERRIDE!
     ! Input                                                    VALUE   !
     !  V001                                                     L001   !
0002 +--[GET]--+---------+---------+---------+---------+---------+--(PUT)--+
     !                                                                    !
     !                                                                    !
     ! Spindle                                                  MAXIMUM !
     ! Override                                                 SPINDLE !
     ! Input                                                    SPEED   !
     !  V002                                                     L006   !
0003 +--[GET]--+---------+---------+---------+---------+---------+--(PUT)--+
     !                                                                    !
     !                                                                    !
     !                                                          AXIS 1    !
     !                                                          PERMIT    !
     !                                                          MOTION    !
     !                                                           N044     !
0004 +---------+---------+---------+---------+---------+---------+--(OTE)--+
     !                                                          !         !
     !                                                          !         !
     !                                                          ! AXIS 2  !
     !                                                          ! PERMIT  !
     !                                                          ! MOTION  !
     !                                                          !  N045   !
     !                                                          +--(OTE)--+
     !                                                                    !
     !                                                                    !
     !                                                          FEEDHOLD!
     !                                                          REQUEST !
     !                                                          INTERNAL!
     !                                                           N025   !
0005 +---------+---------+---------+---------+---------+---------+--(OTD)--+
     !                                                                    !
```

DOCUMENT: 8400GP MIN. PAL EXAMPLE
DATE: 3-7-86



Ladder for Series 8400 Control (continued)



The Minimum Ladder example for the Series 8400 Robot Control is found on page F-36 of Appendix F, immediately following the advanced ladder example.

3.4 Using the Development Program with the Kaypro

This section gives you a series of brief steps for using the development program, PAL.COM and XREFLAD.COM, to generate PAL programs.

The detailed explanation of how to use PAL.COM is given in chapter 5. This outline of the steps is a preview of that chapter.

It is good practice, if this is your first time through 8400 PAL, to use the example program in chapter 4 for your first attempt. It will give you experience with the Kaypro, and help you learn the structure and function of PAL programs.

Development Steps

1. Power up the Kaypro and insert the Main Disk in drive A and the Data Disk in drive B (see chapter 4, Organize Your Development Disks). Type in "PAL" and press [Return]. Press the CAPS LOCK key. PAL.COM only recognizes capital letters.

2. Type "I" to select I/O assignments. You will use parallel, non-inverted I/O and use 7408 buffers for outputs. Set reader or punch enable, if desired. Define each port as input, output, or unused. Define port pairs for V variables, if desired. Type Return to get back to the Main Menu.

3. Type "E" to enter edit mode. Write in the PAL program from chapter 4. Make a list of all messages used by M flags. Reserve status message M028 for the PAL program part number.

4. Once you've input the ladder, assign messages with the message display.

5. Save the PAL source code (PAL source code can be saved at any time and should be saved frequently in case of power loss).

6. Compile the PAL source code and correct any compilation errors. Save the source code again if changes are made. You will have to recompile after saving the changed source code.

7. Download the PAL object code to the 8400 for testing. We strongly recommend that initial testing be done on a test fixture with switches and lights only. Extreme caution is advised when testing on a "live" machine.

8. When the PAL program works properly, save the corrected PAL source code again, and then recompile.

9. Prepare to burn the object code into the EPROMs by the following sequence:

   - If you are burning 2764s for GLC, PAL and for Revision B hardware, type [S] then select 2764. If you are not burning 2764s continue in the sequence in this step.

   - Enter "Hex I/O" mode. Set the baud rate at 1200.

   - Enter terminal mode and initialize the prom programmer.

   - Exit terminal mode and select object code output.

10. Burn object code EPROMs for permanent PAL program installation in the 8400.

11. Make a print out of the PAL program using "PRINT", and make a cross reference of the program using XREFLAD.COM.

4.0 USING PAL.COM and XREFLAD.COM

The first 11 topics in this chapter tell you how to use the PAL.COM program to generate 8400 PAL. The last topic tells you how to use the XREFLAD.COM program to make a cross reference of your PAL program.

Before you use this chapter you should:

- Be familiar with how to operate the Kaypro and CP/M (this is not covered in this manual)
- Know what 8400 PAL is (see chapter1)
- Be familiar with the equipment for 8400 PAL development (see chapter 2)
- Know the programming guidelines and examples we give you (see chapter 3)
- Have your development system organized (see chapter 4.1), especially your development disks.
- Know the various types of PAL instructions (see chapter 5)

In this chapter, when we tell you to press a key on the Kaypro keyboard, we represent the key with its letter, or function, in brackets. For example, if you should press the E key, we show you this with [E].

4.1 Preview of PAL.COM and XREFLAD.COM

PAL.COM and XREFLAD.COM gives you the following functions:

Edit Ladder (the Editor) allows the entry of the 8400 interface program in ladder diagram form.

This function changes the ladder diagram source code that you write on the Kaypro, to object code that can be used on the 8400.

This function loads the ladder diagram object code into the 8400. You use this function to test the ladder before burning it into EPROM.

This function saves the ladder information on a disk. You can use this function at any time while you are using PAL.COM. It should be used to prevent the loss of information before returning to CP/M from the PAL.COM program. It is good practice to save your program frequently. If not saved, you will lose your program if there is a power loss.

This function loads ladder information from a disk file.

This function transfers ladder diagram object code to a prom programmer. It can also be used for any general purpose transfer of ladder information to and from the Kaypro serial port.

The print function gives a print out of the PAL program, I/O assignments and messages.

XREFLAD lets you generate a cross reference of the flags, variables, and rungs in your PAL program.

4.2 Getting Started

Switch on the Kaypro. After a warm up period, you'll see this message on the CRT screen.

```
* KAYPRO *

Please insert your diskette into Drive A
```

Insert a diskette with the CP/M operating system, PAL.COM, and XREFLAD.COM (your Main Disk) into disk drive A and close the disk door. Disk drive A will activate and the CP/M operating system will begin to load into the Kaypro.

After a few seconds you'll see this on the CRT screen.

```
KAYPRO II 64K CP/M V 2.2

A>
```

The default disk drive is now drive A.

Insert your Data Disk into drive B. When you save data to a disk file, you must use the conventions of the CP/M operating system. Refer to the Kaypro manuals for more information on the default disk drive, and file name conventions.

Starting the PAL.COM Program

Type [P] [A] [L] [RETURN]. After PAL.COM loads, you'll see the PAL.COM Main Menu on the CRT screen:

```
ALLEN-BRADLEY LADDER SUPPORT
COPYRIGHT  1985 REV Y28

MAIN MENU
I = I/O ASSIGNMENTS DISPLAY & MODIFY
E = EDIT LADDER
C = COMPILE
D = DOWNLOAD LADDER TO 8400

L = LOAD FROM A DISK FILE
S = SAVE TO A DISK FILE

H = HEX TRANSFER IN & OUT

M = MESSAGES DISPLAY & MODIFY

P = PRINT

HOLD "CTRL" AND TOUCH "C" TO RETURN TO CPM SYSTEM
```

The PAL.COM program implements a menu addressed method of operation. For easy interaction with the program, major functions are divided into "menus" shown on the CRT. Each menu displays information and allows you to interact with a specific function.

Most of the menu functions are activated with one keystroke. This makes your use of a function as easy as possible.

To select any of the PAL.COM functions, type the letter of the function you want.

Important: The [CAPS LOCK] key must be activated before any command will be recognized.

If you return to the CP/M operating system by holding [CTRL] and typing [C], first make sure that you save the PAL program you are working on. See chapter 5.8. If you return to CP/M before you save the PAL program, any PAL program information in PAL.COM will be lost.

You must first initialize (format) the Data Disk in drive B before any data is stored there. The very first save to a disk file on your blank Data Disk in drive B, will not work if the disk has not been initialized.

Steps for Generating a New Ladder

To create a new ladder, the sequence of operations is:

1. Define the 8400 I/O assignments.
2. Define the messages and their text for the PAL program.
3. Use the Edit Ladder functions to create the ladder diagram source code.
4. Save the I/O assignments, ladder diagram, and messages to a disk file.
5. Compile the ladder diagram source code to create the ladder diagram object code.
6. Download the ladder diagram object code to the 8400 for a trial run.

CAUTION: We strongly recommend you test initial PAL programs on a test fixture that simulates the action of inputs and outputs for a real machine. Extreme caution should be used when testing PAL programs on a "live" machine. Unexpected action of machine tool peripheral devices, such as the spindle and tool changer, could result and pose a potential hazard to personnel and equipment. When downloading PAL programs to the 8400 connected to a "live" machine, first establish an Emergency Stop condition, that disables the axis drives.

Repeat steps 1, 2, 3, 4, 5, and 6 until machine operation is satisfactory.

7. Transfer the ladder diagram object code to a prom programmer. Burn an EPROM with the proven program.

8. Insert the EPROM into the 8400 for permanent operation.

9. Print out the I/O assignments, ladder diagram, and messages to a printer.

4.3 I/O Assignments Display and Modify

Press [I] on the Main Menu page to use I/O Assignments Display and Modify.

The I/O assignment process lets you:

- Define the type of interface that is fitted to the 8400 control.
- Choose normal or inverted I/O.
- Define the assignments of PAL ports in the 8400 system as input, output, or unused.
- Dedicate PAL ports to parallel peripheral devices.
- Assign V variables to PAL port two at a time for binary word I/O.

Each PAL program that you develop should have I/O assignments included with it. With some earlier versions of PAL.COM, if you develop a program without I/O assignments, then save this program to a disk file, an error will result when you try to load this program back from the disk file. To recover the saved program, use the Append Source Code option in the Load From a Disk File Menu.

The Type of Interface

All your PAL programs will use the parallel interface. The multiplexed interface requires special hardware for operation that is not offered by Allen-Bradley.

Parallel interface uses the standard 8400 8-bit I/O ports. The operation of the program will use the parallel interface.

When the interface is active the following menu is given on the CRT screen:

```
I/O POLARITY IS NORMAL                  R001 THRU R008 PORT 01 ALLOCATED
                                        R009 THRU R016 PORT 02 ALLOCATED
PARALLEL READER IS NOT ACTIVE           R017 THRU R024 PORT 03 ALLOCATED
         PUNCH IS NOT ACTIVE            R025 THRU R032 PORT 04 ALLOCATED
         CASSETTE IS NOT ACTIVE         R033 THRU R040 PORT 05 ALLOCATED
                                        R041 THRU R048 PORT 06 ALLOCATED
           VAR#          HI    LO       R049 THRU R056 PORT 07 ALLOCATED
           V001 = PORTS                 R057 THRU R064 PORT 08 ALLOCATED
              2                         R065 THRU R072 PORT 09 ALLOCATED
              3                         R073 THRU R080 PORT 10 ALLOCATED
              4                         R081 THRU R088 PORT 11 ALLOCATED
              5                         R089 THRU R096 PORT 12 ALLOCATED
              6                         R097 THRU R104 PORT 13 ALLOCATED
              7                         R105 THRU R112 PORT 14 ALLOCATED
              8                         R113 THRU R120 PORT 15 ALLOCATED

            I/O MENU
           P = PERIPHERALS ACTIVE SELECT
           V = VARIABLE I/O PORT GROUPINGS SELECT
           D = DEFINE PORTS "INPUT" "OUTPUT" "UNUSED"
           R = INVERTED/NON-INVERTED I/O TOGGLE ON/OFF

           "RETURN" = RETURN TO MAIN MENU
```

Choose the Type of I/O

Select inverted or normal I/O next. Choosing inverted or normal I/O changes how the ladder instructions handle real inputs and outputs (flags R001 to R120 and their associated V variables).

The default of the program is inverted I/O.

Important: we recommend you use non-inverted (normal) I/O. If you change from inverted to non-inverted I/O, you should do so before using any of the remaining functions in the I/O Menu.

To change from inverted to normal I/O press [R]. All current I/O assignments will be cancelled by this action.

Now press [Y] to change from inverted to normal I/O, or press [N] to leave inverted I/O in place. The I/O Menu will display the I/O type you have selected.

Choosing inverted or normal I/O does not change how the parallel peripheral devices (i.e. Reader Cassette, Punch) work in any way.

Choose the Type of Peripherals

The next step is to select which peripherals the 8400 will use. To do this, press [P] when you see the I/O Menu. You'll see the following in the lower portion of the CRT screen:

```
    PERIPHERAL MENU
R = READER TOGGLE ON/OFF
P = PUNCH TOGGLE ON/OFF
C = CASSETTE TOGGLE ON/OFF

"RETURN" = RETURN TO I/O MENU
```

Choosing a peripheral active will cause certain PAL ports to be dedicated to a peripheral. When you choose a peripheral to be ON, certain parallel ports cannot be used by PAL for discrete I/O.

This is a list of PAL ports that are dedicated when the peripherals are active:

Connector	Device		Dedicated R Flags and Ports	
CRT-CN5	Punch	=	R097 through R104, port 13	DEDICATED
			R105 through R112, port 14	DEDICATED
			R113 through R120, port 15	DEDICATED
CPU-CN16	Reader	=	R025 through R032, port 04	DEDICATED
			R033 through R040, port 05	DEDICATED
			R041 through R048, port 06	DEDICATED
CPU-CN15 & CPU-CN16	Cassette	=	R001 through R008, port 01	DEDICATED
			R009 through R016, port 02	DEDICATED
			R017 through R024, port 03	DEDICATED
			R025 through R032, port 04	DEDICATED
			R033 through R040, port 05	DEDICATED
			R041 through R048, port 06	DEDICATED

Choose a Reader, Punch, or Cassette by typing [R], [P], or [C]. These functions "toggle." The first time you press [R] the reader will be ON and the second time you press [R] it will be OFF.

After pressing [R], [P], or [C] you'll see the I/O Menu again. The default condition of any peripheral is OFF.

Choose PAL Ports as Input, Output or Unused

The third step is to define which ports are input, output, or unused. A port is always unused if it is not defined as input or output.

Press [D] on the I/O Menu to start this step. You'll see the following in the lower portion of the CRT screen:

```
ENTER TWO DIGIT PORT NUMBER FROM "01 TO 15"

    "RETURN" = RETURN TO I/O MENU
```

Now select which port you want to define by typing a two digit port number.

The following menu will appear in the lower portion of the CRT screen:

```
   DEFINE I/O MENU
I = INPUT SELECT
O = OUTPUT SELECT
U = UNUSED SELECT

"RETURN" = RETURN TO I/O MENU
```

Press [I] for input, [O] for output, or [U] for unused. After pressing [I], [O], or [U] you'll see the I/O Menu again. Repeat these steps to assign each PAL port. The default of any port is unused.

Choose V Variables

The final step defines which PAL ports make up V variables.

You'll see the following in the middle left side of the CRT screen:

```
PARALLEL  I/O POLARITY IS NORMAL
          READER IS NOT ACTIVE
          PUNCH IS NOT ACTIVE
          CASSETTE IS NOT ACTIVE


             VAR#            HI    LO
             V001 = PORTS
                2
                3
                4
                5
                6
                7
                8

 I/O MENU
P = PERIPHERALS ACTIVE SELECT
V = VARIABLE I/O PORT GROUPINGS SELECT
D = DEFINE PORTS "INPUT" "OUTPUT" "UNUSED"
R = INVERTED/NON-INVERTED I/O TOGGLE ON/OFF

"RETURN" = RETURN TO MAIN MENU
```

Since each PAL port is 8-bits wide when the parallel interface is active, two PAL ports need to be entered to define a 16-bit V variable. You enter one port number to make up the upper 8 bits (shown as HI - most significant byte), and one port number to make up the lower 8 bits (shown as LO - least significant byte).

The HI bit of a byte is the bit with the highest R-Flag number and the LO bit of a byte is the bit with the lowest R-Flag number. When you compare two bytes together, one byte will contain the bit with the highest R-Flag number of the two bytes. The byte with the highest R-Flag number is the HI byte.

A look at the I/O assignment sheets in the beginning of chapter 3 will help to illustrate this point.

Assigning V variables to ports does not affect the action of any associated R flags. The R flags can still be used as individual inputs or outputs in the PAL program according to their I/O assignments.

Press [V] while you see the I/O Menu to assign PAL ports to V variables. The lower portion of the CRT screen will display:

```
ENTER VARIABLE NUMBER "1 TO 8" OR TOUCH "RETURN" TO RETURN TO I/O MENU

 V00_
```

After you assign the port as an input or output, define it as a variable port.

Enter a one digit number between 1 and 8 to select the V variable you want. Not all V variables need to be defined.

When a V number is given, the display will request the port numbers to assign. When the parallel interface is active, you'll see this in the lower portion of the CRT screen:

```
ENTER PORT NUMBER FROM "001 TO 015" OR TOUCH "RETURN" TO RETURN TO I/O MENU
```

Important: If you only want to use 8 bits of the V variable, you can assign both the MSB (HI) and the LSB (LO) with the same port number. For example, if you know that only the LSB of V1 will be used, you might assign HI = 08 (port 8), and LO = 08 (port 8 again) to V1. Duplicating the port assigned to MSB and LSB has no effect on the operation of your PAL program.

Finishing I/O Assignments

Once you've chosen the interface, assigned inputs and outputs, and V variables, this finishes the task of I/O assignment. Touch [RETURN] to return to the Main Menu.

4.4 Messages Display and Modify

This function allows you to examine and edit machine-dependent-messages (such as "Door Open", "Coolant On," etc.). To perform this function, press [M] while on the Main Menu. You'll see the following on the CRT screen:

M#	ALARM MESSAGES	M#	STATUS MESSAGES	M#	STATUS MESSAGE
001	M001 ALARM MESSAGE	014	M014 STATUS	029	M029 STATUS
002	M002 ALARM MESSAGE	015	M015 STATUS	030	M030 STATUS
003	M003 ALARM MESSAGE	016	M016 STATUS	031	M031 STATUS
004	M004 ALARM MESSAGE	017	M017 STATUS	032	M032 STATUS
005	M005 ALARM MESSAGE	018	M018 STATUS	033	M033 STATUS
006	M006 ALARM MESSAGE	019	M019 STATUS	034	M034 STATUS
007	M007 ALARM MESSAGE	020	M020 STATUS	035	M035 STATUS
008	M008 ALARM MESSAGE	021	M021 STATUS	036	M036 STATUS
009	M009 ALARM MESSAGE	022	M022 STATUS		
		023	M023 STATUS		
M#	STATUS MESSAGES	024	M024 STATUS	M#	VARIABLE MESSAGES
010	M010 STATUS	025	M025 STATUS	037	M037 VARBL
011	M011 STATUS	026	M026 STATUS	038	M038 VARBL
012	M012 STATUS	027	M027 STATUS	039	M039 VARBL
013	M013 STATUS	028	M028 STATUS	040	M040 VARBL

WHAT MESSAGE DO YOU WISH TO ENTER OR MODIFY?

ENTER 2 DIGIT NUMBER BETWEEN 001 AND 040 OR "RETURN" TO RETURN TO MAIN MENU

MESSAGE M_

There are a total of 40 messages. The first 9 are alarm messages. The next 27 are status messages and the last 4 are variable messages.

Alarm Messages

The alarm message flags, M001 to M009, when set by an instruction in the ladder show their associated messages on the 8400 CRT. The text of an alarm message can be up to 20 characters in length.

Important: The 8400 will be forced into a FEEDHOLD condition when an alarm message flag is set.

Only one alarm message is displayed at a time on the 8400. If the ladder enables more than one, the alarm message with the lowest "M" number (the highest priority) will be displayed.

Status Messages

The status message flags, M010 to M036, when set by an instruction in the ladder, display their assigned messages on the 8400 CRT. The text of a status message can be up to 18 characters in length.

The 8400 will NOT be forced into a feedhold condition when a status message flag is set.

Up to eight status messages can be displayed at one time on the 8400 status page. If the ladder sets more than eight, the status messages with the lowest "M" numbers (the highest priorities) will be displayed.

Variable Messages

The variable message flags, M037 to M040, when set by an instruction in the ladder, will display their assigned messages on the 8400 CRT. The text of a variable message can be up to 11 characters in length.

The 8400 will NOT be forced into a FEEDHOLD condition when a variable message is set. The current value in the L variable associated with the flag will be shown on the 8400 CRT along with the text of the message. The assignments of L variables for each variable message flag are as follows:

Variable Message Number	Arithmetic Variable Number
M037	L037
M038	L038
M039	L039
M040	L040

Up to two variable messages can be displayed at one time on the 8400 status page. If the ladder sets more than two variable message flags, the variable messages with the lowest "M" numbers (the highest priorities) will be displayed.

Editing a Message

To edit a message, type the 2 digit number of the flag whose message is to be modified (or enter one or two digits and hit [RETURN]). For example, if [1] [7] [RETURN] is typed, the following will appear in the lower portion of the CRT screen:

```
              MESSAGE M017
TYPE "Y" FOR YES, "N" FOR NO, OR TOUCH "RETURN" IF NO CHANGE IS WANTED
              INVERSE VIDEO ?
```

Select if the message should be shown with inverse video by pressing [Y] for yes, [N] for no, or [RETURN] to leave it unchanged. Assuming [Y] is pressed, you'll see the following:

```
MESSAGE M017
INVERSE VIDEO Y
BLINK ?

TYPE "Y" FOR YES, "N" FOR NO, OR TOUCH "RETURN" IF NO CHANGE IS WANTED
```

Select if the message is to blink by pressing [Y] for yes, [N] for no, or [RETURN] to leave to leave it unchanged. If blink is selected, the message will flash on and off at about a 2 Hz. rate when it is shown on the 8400 CRT. Assuming [Y] is pressed, you'll see the following on the CRT screen:

```
MESSAGE M017
STATUS MESSAGE MAXIMUM LENGTH IS 18 CHARACTERS
ENTER MESSAGE AND "RETURN" OR TOUCH "RETURN" IF NO CHANGES REQUIRED
[                  ]
```

Type the text of the message or press [RETURN] before any text is entered if no change in the message is required. The maximum number of characters allowed is indicated on the display for each type of message. Press [RETURN] when the message is complete. If you type more than the maximum number of characters for a message, only the maximum number of characters will be used.

If an error is made while entering the text, pressing [BACKSPACE] will allow backing the cursor up.

Assuming the text "LOW LUBE LEVEL" is entered, the message M017 will appear as follows on the CRT screen:

```
M#   ALARM MESSAGES        M#    STATUS MESSAGES     M#   STATUS MESSAGE
001  M001 ALARM MESSAGE    014   M014 STATUS         029  M029 STATUS
002  M002 ALARM MESSAGE    015   M015 STATUS         030  M030 STATUS
003  M003 ALARM MESSAGE    016   M016 STATUS         031  M031 STATUS
004  M004 ALARM MESSAGE    017IB LOW LUBE LEVEL      032  M032 STATUS
005  M005 ALARM MESSAGE    018   M018 STATUS         033  M033 STATUS
006  M006 ALARM MESSAGE    019   M019 STATUS         034  M034 STATUS
007  M007 ALARM MESSAGE    020   M020 STATUS         035  M035 STATUS
008  M008 ALARM MESSAGE    021   M021 STATUS         036  M036 STATUS
009  M009 ALARM MESSAGE    022   M022 STATUS
                           023   M023 STATUS
M#   STATUS MESSAGES       024   M024 STATUS         M#   VARIABLE MESSAGES
010  M010 STATUS           025   M025 STATUS         037  M037 VARBL
011  M011 STATUS           026   M026 STATUS         038  M038 VARBL
012  M012 STATUS           027   M027 STATUS         039  M039 VARBL
013  M013 STATUS           028   M028 STATUS         040  M040 VARBL

WHAT MESSAGE DO YOU WISH TO ENTER OR MODIFY?

ENTER 3 DIGIT NUMBER BETWEEN 001 AND 040 OR "RETURN" TO RETURN TO MAIN MENU
      MESSAGE M0_
```

The inverse video and blink options are indicated by an "I" and "B" beside the message number, if selected. Use the above procedure to edit each message you want to enter.

When you finish editing messages, touch [RETURN] to return to the Main Menu.

4.5 Edit Ladder

Press [E] on the Main Menu to select this function.

Edit Ladder allows you to enter, examine and modify PAL programs. When you select Edit Ladder for the first time, you'll see the following on the CRT screen:

```
-E-N-D-+                                        EDIT MENU
                                                I = INSERT
                                                D = DELETE
                                                N = NEXT
                                                L = LAST
                                                S = SEARCH
                                                A = SEARCH AGAIN
                                                "RETURN" =
                                                    MAIN MENU

                    INPUTS                                      OUTPUTS
01--] [--   02--]/[--   80-(LBL)-   40-(GET)-   10-(OTE)-   11-(OTD)-
50-(GTI)-   60-(DGT)-   41-(ADD)-   51-(ADI)-   12-(OTL)-   13-(OTU)-
42-(SUB)-   52-(SBI)-   43-(MUL)-   53-(MLI)-   89-(GTO)-   20-(TON)-
44-(DIV)-   54-(DVI)-   45-(EQL)-   55-(EQI)-   21-(TOF)-   22-(RTO)-
46-(LES)-   56-(LSI)-   70-(SLL)-   71-(SLR)-   23-(RTR)-   30-(CTU)-
78-(DTB)-   79-(BTD)-   00------               31-(CTD)-   32-(CTR)-
90-BRANCH BEGIN         99-BRANCH END          49-(PUT)-   69-(DPT)-
```

Important: Allow the screen to update completely before typing any keys on the Kaypro. The Kaypro does not have a "type-ahead" buffer so keystrokes made while the Kaypro is busy are lost.

The -E-N-D-+ sequence is displayed in the upper left corner of the CRT screen. At this point, this indicates that there are no rungs in the PAL program.

At the bottom of the screen, all the ladder instruction codes are given for reference. The upper right portion of the screen displays the Edit Menu options.

4.5.1 Insert

To start entering a rung, press [I] for insert while you see the -E-N-D-+ sequence and the Edit Menu. You'll see the following screen showing an empty ladder framework, and the Insert Mode in the upper right portion of the CRT screen:

```
    !                                                  !
0001!                                                  !INSERT MODE
    !                                          +       !
    !                                                  !ENTER ELEMENT
    !                                                  ! "ESC" =
    !                                          +       !EDIT MENU
    !                                                  !
    !                                                  !
    !                                          +       !
    !                                                  !
    !                                                  !
    !                                          +       !
    !                                                  !
    !                                                  !
    !                                          +       !
    !                                                  !
ELEMENT --
                    INPUTS                                        OUTPUTS
01--] [--   02--]/[--   80-(LBL)-   40-(GET)-   10-(OTE)-   11-(OTD)-
50-(GTI)-   60-(DGT)-   41-(ADD)-   51-(ADI)-   12-(OTL)-   13-(OTU)-
42-(SUB)-   52-(SBI)-   43-(MUL)-   53-(MLI)-   89-(GTO)-   20-(TON)-
44-(DIV)-   54-(DVI)-   45-(EQL)-   55-(EQI)-   21-(TOF)-   22-(RTO)-
46-(LES)-   56-(LSI)-   70-(SLL)-   71-(SLR)-   23-(RTR)-   30-(CTU)-
78-(DTB)-   79-(BTD)-   00------               31-(CTD)-   32-(CTR)-
90-BRANCH BEGIN         99-BRANCH END          49-(PUT)-   69-(DPT)-
```

Type the two digit code for the instruction (ELEMENT --) you want. The instruction you choose will be shown. If you want to keep the instruction in the PAL program, press [RETURN]. Press any other key to remove it.

Next, the screen will prompt for the instruction's flag or variable (VARIABLE --). Type in the desired flag or variable. To backspace your entry, use the [DELETE] key. Pressing [DELETE] when the entry is full will delete the entire entry. If you want to keep the flag or variable, press [RETURN].

The system will display only one rung at a time.

Continue entering instructions as necessary.

Important: All PAL program rungs must end with at least one output instruction.

Once you enter an output instruction, press [RETURN] to complete the rung. You will automatically return to the Edit Menu.

- To start a new rung at the end of the program, press [I] while you see the -E-N-D-+ sequence and the Edit Menu.

- To insert a new rung among existing rungs, move up or down the ladder using the [N] (next) and [L] (last) keys until you see the rung you want to follow the new rung. Press [I] for insert and continue as above.

Rungs are automatically numbered in the left-hand column. Inserting or deleting rungs cause the rungs that follow to be renumbered.

You can leave the the insert function of the Editor by pressing the [ESC] key when the Editor is prompting for an instruction code. This will exit the insert function, and still leave you in the Editor. The rung that was being inserted when the [ESC] key is pressed will not be entered into the ladder source code.

Important: If you have to change one instruction in a rung, you must delete the entire rung and then insert the rung with the corrected instruction. It is easier to change a rung by entering the new rung before you delete the old one. Here are some points to remember when you edit PAL.

- Instructions must be in proper sequence and order.

- Variables must be of the proper type.

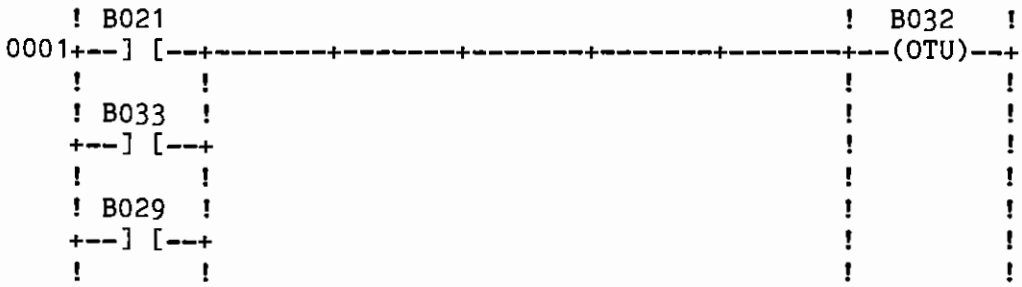
- I/O (R-word) must be assigned before it can be used.

- Edits are inserted before the rung.

When the rung structure requires the use of Branch Start and Branch End functions (codes 90 and 99), the Editor requires the entry of instructions according to the structure. Each rung is limited to (7) seven instructions across (i.e. six input instructions and one output instruction) and (5) five instructions down.

You have to enter a Branch End when you complete a series of parallel sub-rungs (an input group) even though the Branch End is already displayed. This tells the Editor to move back up to the top of the rung.

This is a practice for entering parallel rungs; enter the following commands:



Step 1. Type [9] [0] [RETURN]..... = Branch Begin

Step 2. Type [0] [1] [RETURN]..... = +--] [--+

Step 3. Type [B] [0] [2] [1] [RETURN]..... = flag B021

Step 4. Type [9] [9] [RETURN]..... = Branch End

Step 5. Type [9] [0] [RETURN]..... = Branch Begin

Step 6. Type [0] [1] [RETURN]..... = +--] [--+

Step 7. Type [B] [0] [3] [3] [RETURN]..... = flag B033

Step 8. Type [9] [9] [RETURN]..... = Branch End

Step 9. Type [0] [1] [RETURN]..... = +--] [--+

Step 10. Type [B] [0] [2] [9] [RETURN]..... = flag B029

Step 11. Type [9] [9] [RETURN]..... = Branch End (return to top rung)

Step 12. Type [1] [3] [RETURN]..... = +-(OTU)-+

Step 13. Type [B] [0] [3] [2] [RETURN]..... = flag B032

Step 14. Press [RETURN]..... = Exit Insert Mode and return to Edit Menu

Press [RETURN] once more to exit to the Main Menu.

4.5.2 Delete

The delete function can be used to delete just one rung, or all the rungs in a ladder.

To Delete One Rung in a Ladder

To delete one rung, move up the ladder with [L] (last rung) or down the ladder with [N] (next rung) until the rung you want to delete is displayed.

Assume that rung 0001 from our previous example should be deleted. Move the ladder until rung 0001 is on the CRT screen. Now press [D] and you'll see the following on the CRT screen (assuming you have the example rung in the ladder):

```
    ! B021                                          !  B032   !
0001+--] [--+------+------+------+------+------+--(OTU)--+ DELETE MODE
    !       !                                      !        !
    ! B033  !                                      !        !
    +--] [--+                                      !        !
    !       !                                      !        !ANY KEY =
    ! B029  !                                      !        ! EDIT MENU
    +--] [--+                                      !        !
    !       !                                      !        !
    !       !                                      !        !
    !       !                                      !        !
    !       !                                      !        !
    !       !                                      !        !
    !       !                                      !        !
    !       !                                      !        !
PRESS "RETURN" TO DELETE SEQ.
INPUTS                                                      OUTPUTS
01--] [--   02--]/[--   80-(LBL)-   40-(GET)-   10-(OTE)-   11-(OTD)-
50-(GTI)-   60-(DGT)-   41-(ADD)-   51-(ADI)-   12-(OTL)-   13-(OTU)-
42-(SUB)-   52-(SBI)-   43-(MUL)-   53-(MLI)-   89-(GTO)-   20-(TON)-
44-(DIV)-   54-(DVI)-   45-(EQL)-   55-(EQI)-   21-(TOF)-   22-(RTO)-
46-(LES)-   56-(LSI)-   70-(SLL)-   71-(SLR)-   23-(RTR)-   30-(CTU)-
78-(DTB)-   79-(BTD)-   00-----                 31-(CTD)-   32-(CTR)-
90-BRANCH BEGIN         99-BRANCH END           49-(PUT)-   69-(DPT)-
```

Press [RETURN] to delete this rung, or press any other key on the keyboard to leave the rung in place. If you leave the rung intact, you'll see the Edit Menu again on the CRT screen.

To Delete All Rungs in a Ladder

To delete all the rungs in a ladder, move down the ladder until -E-N-D-+ is displayed. Press [D] and you'll see the following on the CRT screen:

```
-E-N-D-+                                                DELETE MODE


                                                        PRESS ANY KEY
                                                        TO RETURN TO
                                                        EDIT MENU


PRESS "ESC" TO DELETE LADDER
                    INPUTS                                      OUTPUTS
01--] [--   02--]/[--   80-(LBL)-   40-(GET)-   10-(OTE)-   11-(OTD)-
50-(GTI)-   60-(DGT)-   41-(ADD)-   51-(ADI)-   12-(OTL)-   13-(OTU)-
42-(SUB)-   52-(SBI)-   43-(MUL)-   53-(MLI)-   89-(GTO)-   20-(TON)-
44-(DIV)-   54-(DVI)-   45-(EQL)-   55-(EQI)-   21-(TOF)-   22-(RTO)-
46-(LES)-   56-(LSI)-   70-(SLL)-   71-(SLR)-   23-(RTR)-   30-(CTU)-
78-(DTB)-   79-(BTD)-   00------               31-(CTD)-   32-(CTR)-
90-BRANCH BEGIN         99-BRANCH END          49-(PUT)-   69-(DRT)-
```

Press [ESC] to delete all the rungs in the ladder. Press any other key to leave the ladder in place. If you don't delete the ladder, you'll see the Edit Menu again on the CRT screen. Deleting all the rungs in a ladder does not change the I/O assignments or the ladder messages.

4.5.3 Search

The search function is used to find variables or flags that appear in output instructions.

The search function cannot be used to find rung numbers, or label numbers, or flags or variables that are parts of input instructions.

Press [S] when the Edit Menu is displayed and the you'll see the last rung you were working with on the CRT screen (we assume you still have our example rung in the ladder):

```
    ! B021                                          !  B032   !
0001+--] [--+-------+------+------+------+-------+--(OTU)--+ SEARCH MODE
    !       !                                      !         !
    ! B033  !                                      !         !
    +--] [--+                                      !         !
    !       !                                      !         !`DELETE' TO
    ! B029  !                                      !         !  BACKSPACE
    +--] [--+                                      !         !  ENTRY
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
    !       !                                      !         !
VARIABLE?
                        INPUTS                                  OUTPUTS
01--] [--   02--]/[--   80-(LBL)-   40-(GET)-   10-(OTE)-   11-(OTD)-
50-(GTI)-   60-(DGT)-   41-(ADD)-   51-(ADI)-   12-(OTL)-   13-(OTU)-
42-(SUB)-   52-(SBI)-   43-(MUL)-   53-(MLI)-   89-(GTO)-   20-(TON)-
44-(DIV)-   54-(DVI)-   45-(EQL)-   55-(EQI)-   21-(TOF)-   22-(RTO)-
46-(LES)-   56-(LSI)-   70-(SLL)-   71-(SLR)-   23-(RTR)-   30-(CTU)-
78-(DTB)-   79-(BTD)-   00------               31-(CTD)-   32-(CTR)-
90-BRANCH BEGIN         99-BRANCH END          49-(PUT)-   69-(DPT)-
```

Type the number of the flag or variable you want to find and press [RETURN]. The search function will find and display the first rung that contains the flag or variable in an output instruction. When the search stops, you'll see the Edit Menu again.

If the flag or variable is not found in an output instruction, the message "NOT FOUND" will appear on the CRT screen. If this occurs, press any key to return to the Edit Menu.

4.5.4 Faults During Edit Ladder

The Editor detects and reports invalid entries and commands. It displays error messages when you make these mistakes. Table 5-1 at the end of this section

lists the error codes that you may encounter, and the possible reasons for an error.

4.5.5 Exit Editor

To exit the Edit Ladder function, press [RETURN] while the Edit Menu is displayed. This brings the Main Menu back to the CRT screen. The Editor cannot be exited while you are inserting rungs or searching for flags and variables.

4.6 Compile

Press [C] on the Main Menu page to begin compiling.

The purpose of compile is to change the ladder diagram source code into object code. Any time I/O assignments ladder rungs, or messages are changed, or the program is saved, loaded, or printed, the compile function must be used to get new object code. Changes made to the I/O assignments ladder rungs, or messages will not be included in the object code until the compile function is used. When you compile you'll see the following on the CRT screen:

```
COMPILING . . . .
```

and after a few seconds,

```
COMPILING COMPLETE  00 ERRORS
PRESS [RETURN] TO RETURN TO MAIN MENU
```

You can stop the compile function at any time by pressing the [ESC] key.

Possible Errors

If any errors occur during compiling, there are errors in the ladder structure (see table 5-1 at the end of this section). You'll see error message numbers and the number of the faulty rung(s). You should use Edit Ladder to correct the ladder and compile the ladder again.

Examples of compilation errors are:

```
COMPILING . . . ERROR 02  SEQ 0000
COMPILATION ABORTED.  01 ERRORS
    PRESS RETURN TO RETURN TO MAIN MENU
```

Error 02 states that no ladder rungs exist.

```
COMPILING . . . ERROR 15  SEQ 0004
                ERROR 15  SEQ 0004
                ERROR 15  SEQ 0005
                ERROR 15  SEQ 0006
                ERROR 15  SEQ 0006
                ERROR 15  SEQ 0006
COMPILATION ABORTED.  06 ERRORS
   PRESS RETURN TO RETURN TO MAIN MENU
```

Error 15 states that an R flag in the rungs listed has not been assigned for use by the ladder.

4.7 Download Ladder to 8400

After a ladder has been entered and compiled, you can download it to the 8400 CNC for temporary evaluation and testing.

CAUTION: We strongly recommend you test initial PAL programs on a Test fixture that simulates the action of inputs and outputs for a real machine. Extreme caution should be used when testing PAL programs on a "live" machine. Unexpected action of machine tool peripheral devices, such as the spindle and tool changer, could result and pose a potential hazard to personnel and equipment. When downloading PAL programs to the 8400 connected to a "live" machine, first establish an Emergency Stop condition, that disables the axis drives.

The serial interface cable (8400-XPC3) should be connected between the Kaypro's connector J4 and the 8400 CPU connector CN14 on the CPU board.

On the Robot control, (8400R) the serial interface cable (8400-XPC3) is connected to the Teach Pendant connector and switch 3-4 must be toggled. Then the processor must be reset.

If the 8400 Peripheral Interface Module is used, the cable should be connected to the Communication Port Connector instead of the 8400 CPU Connector CN14.

In either case, the serial port on the 8400 CPU module must be configured for RS-232 operation. JP14 and JP16 on the 8400 CPU Module must be installed for RS-232 operation.

The compiled ladder object code is loaded into the 8400 part program storage RAM. It is good practice to delete programs from program storage to make room for PAL before downloading. The maximum length that any PAL program can have is 12K bytes. After the download is complete, it is good practice to examine the part program storage page to determine how much room is still available.

To Begin Downloading

The 8400 most be in the Jog mode with Feedhold active to download to it. Press [D] when on the Main Menu page to begin the download function.

If a response is received from the 8400 after you press [D], you'll see the following on the Kaypro CRT screen:

```
DOWNLOADING...XX
```

where XX counts up in decimal to indicate a download is in progress.

During downloading, the 8400 CRT will have PAL DISABLED displayed on a reverse video field located just below the Jog and Feedhold messages. The K-1 relay will drop out and the axis will be disabled.

Possible Errors

If the ladder source code has not been compiled, the following message is displayed on the CRT screen:

```
INVALID OBJECT

COMPILATION REQUIRED

TOUCH "RETURN" TO RETURN TO MAIN MENU
```

If the serial interface cable is missing or connections incomplete, the following message will appear on the CRT screen:

```
DOWNLOADING
RESPONSE WAS NOT RECEIVED FROM SYSTEM

TOUCH "RETURN" TO RETURN TO MAIN MENU
```

If transmission stops during the download (for example, the cable connection is broken) the following message will be displayed on the 8400 CRT screen:

MICRO PORT FAILURE PART PROGRAM STORAGE RAM OVERFLOW
PAL DISABLED

TOUCH "RETURN" TO RETURN TO MAIN MENU

If the ladder object code is too large for the 8400 part program RAM, the following message will appear on the CRT screen:

MICRO PORT FAILURE PART PROGRAM STORAGE RAM OVERFLOW
PAL DISABLED

8400 LADDER RAM OVERFLOW

Important: When the 8400 has micro-port failure it must be reset to continue.

Successful Transfer and Start Execution

A successful transfer is indicated by the following message on the Kaypro CRT screen:

DOWNLOADING XX

TRANSFER COMPLETED AND EXECUTING

TOUCH "RETURN" TO RETURN TO MAIN MENU

The DISABLED PAL message goes away from the 8400 CRT upon successful completion of a download.

Important: Touching [RETURN] does not abort PAL execution in the 8400. It simply returns you to the Main Menu. Pressing [RETURN] has no effect on the 8400 nor on RAM based PAL execution. RAM PAL will execute until 8400 power is cycled or until the reset button on the 8400 Control Package is pressed.

Important: Do not leave PAL in RAM. When testing is complete you must burn PAL into EPROM. Leaving PAL in RAM does not provide secure storage for the PAL program. When power to the 8400 is turned off, then on, the PAL program will be lost.

4.8 Save to a Disk File

After the ladder source code has been entered with Edit Ladder, it can be saved to a disk file. It is good practice to save the source code on two disks. Use one for back-up if the first fails. It is also good practice to perform a save function often during lengthy edits. This prevents a power failure from destroying an edit session.

Important: If you exit PAL.COM and go to the CP/M operating system, any active PAL program (one that is in the PAL.COM program) is lost. So, you should save any active program to a disk file before exiting PAL.COM. Press [S] when on the Main Menu page to save to a disk file. The following message will appear on the CRT screen:

```
SAVE TO A DISK FILE

        MENU
S = SOURCE CODE AND MESSAGES SAVE
M = MESSAGES ONLY SAVE

"RETURN" = RETURN TO MAIN MENU
```

Source Code and Messages Save

Press [S] when on the Save to a Disk File Menu to select this function.

You'll see that the program requests the file name. Enter a name for your ladder. All the rules that apply to CP/M disk file name conventions apply here, including the drive name extension (SRC = source code and OBJ = object code). For example,

```
FILE NAME   B:CUSTOMER.SRC
```

If the transfer is completed successfully, you'll see the following message on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
FILE TRANSFER COMPLETED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Important: A second save to the same file name erases the earlier file and saves the active file. No back-up file will be made unless you save to two different files, i.e., CUSTOMER.SRC and CUSTOMER.BAK. We recommend making back-up files. Changes to your programs may not always result in progress! Being able to recover a tested original program is desirable.

Important: The Data disk in drive B must be initialized (formatted) or the very first save to a disk file, on your blank Data Disk in drive B, may not work. You'll have to return to CP/M, and information will then be lost.

Possible Errors

If there isn't any room on the disk, you'll see the following message on the CRT screen:

```
FILE NAME    B:CUSTOMER.SCR
DISK FILE FULL, COMMAND ABORTED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

If the directory is full on the disk, you'll see the following message on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
NO DIRECTORY SPACE, COMMAND ABORTED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Refer to the Kaypro operating manual to find out how to delete unnecessary files from a disk.

Messages Only Save

Press [M] when on the Save to a Disk File Menu to select this function.

You'll see that the program requests the file name. Enter an appropriate name for a message file. All the rules that apply to CP/M disk file name conventions apply here. For example,

```
FILE NAME    B:CUSTOMER.MSG
```

If the transfer completes successfully, the following message will appear on the CRT screen:

```
FILE NAME    B:CUSTOMER.MSG
FILE TRANSFER COMPLETED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Possible Errors

If there isn't any room on the disk, the following message will appear on the CRT screen:

```
FILE NAME    B:CUSTOMER.MSG
DISK FILE FULL, COMMAND ABORTED

TOUCH ANY KEY TO RETURN TO THE MAIN MENU
```

If the directory is full on the disk, the following message will appear on the CRT screen:

```
FILE NAME    B:CUSTOMER.MSG
NO DIRECTORY SPACE, COMMAND ABORTED

TOUCH ANY KEY TO RETURN TO THE MAIN MENU
```

Refer to the Kaypro operating manual to find out how to delete unnecessary files from a disk.

4.9 Load From a Disk File

Selecting Load from a Disk File will allow information that you've saved on disk to be reloaded into the Kaypro.

Press [L] when on the Main Menu page to select this function. The following menu will appear on the CRT screen:

```
LOAD FROM A DISK FILE

                MENU
S = SOURCE CODE FILE LOAD (WITH MESSAGES IF PRESENT)
A = APPEND A SOURCE CODE FILE TO EXISTING SOURCE CODE
M = MESSAGE FILE ONLY LOAD

"RETURN" = RETURN TO MAIN MENU
```

Source Code File Load (With Messages if Present)

Press [S] on the Load from Disk File Menu to select this function.

You'll see that the program requests the file name. Enter the name of the ladder source code file. Be sure to include the drive letter if required. All the rules that apply to CP/M disk file name conventions apply here. For example,

```
FILE NAME    B:CUSTOMER.SRC
```

Important: Any existing ladder source code contained in the Kaypro memory will be erased by this command.

Ladder source code files should contain I/O assignments, ladder, and messages information. If the transfer completes successfully, you'll see the following message on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
FILE TRANSFER COMPLETED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

If a file name is given that does not exist on the disk, the following error message will be displayed on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
FILE NAME SPECIFIED, NOT FOUND ON DISK

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Append a Source Code File to Existing Source Code

Press [A] on the Load from Disk File Menu to select this function.

This function allows ladder rungs from a source code file to be added to the end of any existing rungs in the Kaypro memory.

Appending rungs does not affect the I/O assignments or messages of the active PAL program in the Kaypro. The I/O assignments and messages of the source file that is being added are ignored.

This function allows you to make a library of "building block" rungs. These "building block" rungs could each contain information for one particular function of a standard machine interface: such as tool changer, spindle controller, limit switch information, etc.

You'll see that the program requests the file name. Enter the name of the ladder source code file. Be sure to include the drive letter if required. All the rules that apply to CP/M disk file name conventions apply here. For example,

```
FILE NAME    B:CUSTOMER.SRC
```

If the transfer is completed successfully, the following message will appear on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
FILE TRANSFER COMPLETED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

You can touch any key on the Kaypro keyboard to return to the Main Menu page.

Possible Error

If a file name is given that does not exist on the disk, the following error message is shown on the CRT screen:

```
FILE NAME    B:CUSTOMER.SRC
FILE NAME SPECIFIED, NOT FOUND ON DISK

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Message File Only Load

Press [M] on the Load from a Disk File Menu to select this function.

This function allows you to load a file that has messages only. This function does not affect the source code, or I/O assignments that are present in the Kaypro.

The program will prompt you for the file name. Enter the name of the ladder message file. Be sure to include the drive letter if required. All the rules that apply to CP/M disk file name conventions apply here. For example,

```
FILE NAME    B:CUSTOMER.MSG
```

If the transfer completes successfully, you'll see the following message on the CRT screen:

```
FILE NAME    B:CUSTOMER.MSG
FILE TRANSFER COMPLETED

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

Possible Error

If a file name does not exist on the disk, the following error message is displayed on the CRT screen:

```
FILE NAME    B:CUSTOMER.MSG
FILE NAME SPECIFIED, NOT FOUND ON DISK

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

You can only touch the [RETURN] key on the Kaypro keyboard to return to the Main Menu page. Examine the directory of the disk, and make sure to specify the name of the file correctly.

4.10 Hex Transfer In & Out

The main purpose of Hex Transfer In & Out is to transfer object code to a prom programmer. You want to create an EPROM resident PAL program for permanent storage in the 8400 control.

Hex Transfer In & Out also allows:

- transfering source code and messages to and from the Kaypro serial port. This function can be used to transfer information to another storage device or computer.
- the Kaypro to act as a terminal. This is useful for performing special functions with the prom programmer.

To select Hex Transfer In & Out, touch [H] while in the Main Menu. The following menu will appear on the CRT screen:

```
HEX TRANSFER INPUT & OUTPUT                I/O MODE IS: HALF DUPLEX
          MENU
       I = INPUT HEX
       O = OUTPUT HEX
       B = BAUD RATE SELECT
       T = TERMINAL MODE
       F = FULL DUPLEX
       H = HALF DUPLEX

'RETURN' = RETURN TO THE MAIN MENU
```

You can only touch the [RETURN] key on the Kaypro keyboard to return to the Main Menu page.

Important: The first time you use the Hex Transfer Input & Output Menu, set the baud rate before going on to the next function.

Baud Rate Selection

Press [B] on the Hex Transfer Input & Output Menu to select this function. The following appears on the CRT screen:

```
   50 BAUD-"0"
   75 BAUD-"1"
  100 BAUD-"2"
  134 BAUD-"3"
  150 BAUD-"4"
  300 BAUD-"5"            SELECT THE DESIRED BAUD RATE BY
  600 BAUD-"6"            PRESSING THE KEY LISTED NEXT TO
 1200 BAUD-"7"                 THE BAUD RATE REQUIRED.
 1800 BAUD-"8"
 2000 BAUD-"9"
 2400 BAUD-"A"
 3600 BAUD-"B"     "RETURN" =  RETURN TO MAIN MENU
 4800 BAUD-"C"
 7200 BAUD-"D"
 9600 BAUD-"E"
19200 BAUD-"F"
```

Select the baud rate by pressing the key that is indicated next to the baud rate. For example, if you want 1200 baud, press [7]. You'll automatically return to the Hex Transfer Input & Output Menu once you choose the baud rate. Pressing [RETURN] is not required.

The baud rate you choose will remain in effect until it is changed, or until the downloading function is used. Downloading will change the baud rate to 1200 baud.

The default rate is 1200 baud, and 1200 baud is recommended for Bay Technical Associates prom programmers.

I/O Mode Selection

Before proceeding with output, select the I/O mode

- full duplex
- half duplex

Half-duplex is the default.

This step sets up the Bay Tech. for the type of chip you are going to burn.

If the prom programmer echoes characters, use full duplex; if no echo use half duplex. Full duplex is recommended for Bay Technical Associates prom programmers. Handshaking must always be present.

4.10.1 Output Hex

Press [O] on the Hex Transfer Input & Output Menu to select the output hex function. You'll see the following on the CRT screen:

```
HEX TRANSFER OUTPUT

              MENU
S = SOURCE CODE OUTPUT WITH MESSAGES
O = OBJECT CODE OUTPUT
M = MESSAGES ONLY OUTPUT

TOUCH 'RETURN' TO RETURN TO THE MAIN MENU
```

You can only touch the [RETURN] key on the Kaypro keyboard to return to the Main Menu page.

Object Code Output

Press [O] on the Hex Transfer Output Menu to select this function.

You use this function to output object code from the Kaypro serial port for burning into EPROM. The object code will become the EPROM resident PAL program in the 8400.

The maximum length of a PAL program is 12K bytes. So, if you have an 8400 L, GP, MP, or Bandit III there is a maximum of 6 EPROMs to burn (because they use 2716's). If you have an 8400 GLC there is a maximum of 2 EPROMs to burn (because it uses 2764's).

Serial output of the object code from the Kaypro is in Intel Hex format (see table 2 at the end of this section for a description of Intel Hex format).

Possible Error

The ladder source code must be compiled into object code before it can be output from serial port on the Kaypro.

If you don't compile before you output, the system recognizes this and you'll see the following message:

INVALID OBJECT

COMPILATION REQUIRED

"RETURN" = RETURN TO THE MAIN MENU

See chapter 5.6 for how to compile the ladder source code into object code.

Choosing EPROM Size

As a reminder for choosing the right EPROM size this menu appears on the screen:

This step tells the prom burner the size of chip you are going to burn.

SELECT PROM SIZE

2716 - '1'
2732 - '2'
2764 - '3'
27128 - '4'

OR TOUCH "RETURN" TO RETURN TO MAIN MENU

The program will output a certain amount of object code at one time depending on the EPROM size you choose. Select the EPROM size by pressing [1], [2], or [3] on the keyboard.

Currently, the 8400 GP, R, MP, Bandit III and Bandit IV use 2716 EPROM's, and the 8400 GLC uses 2764's.

- Pressing [1] (selecting 2716) will transfer 2048 bytes at one time.
- Pressing [2] (selecting 2732) will transfer 4096 bytes at one time.
- Pressing [3] (selecting 2764) will transfer 8192 bytes at one time.
- Pressing [4] (selecting 27128) will transfer 16384 bytes at one time.

Initialize the PROM Programmer

Before starting the transfer, you must initialize the programmer and prepare it for receiving Intel Hex format. Some programmers require certain commands to allow them to receive Intel Hex format. Other programmers have keyboards which allow you to enable Intel Hex format directly.

Each prom programmer has its own way to enable Intel Hex format. Refer to your prom programmer manual to determine what is needed here.

The menu tells you that any Kaypro keyboard entry will be transferred to the serial port. You can use this to prepare the programmer for Intel Hex format, if required. The Kaypro will be in terminal mode until you hold [CTRL] an press [S] to begin transfer.

After you choose the EPROM size, the Object Output Ready Menu appears on the CRT screen:

```
OBJECT OUTPUT READY

KEYBOARD ENTRIES ARE NOW TRANSFERED TO SERIAL PORT.

TYPE WHATEVER IS NEEDED (CONSULT YOUR PROM PROGRAMMERS MANUAL)
TO PROMPT THE PROGRAMMER TO RECEIVE INTEL HEX CODE FROM THE KAYPRO.

HOLD 'CRTL' AND TOUCH 'S' TO START TRANSFER

"ESC" = ABORT AND RETURN TO MAIN MENU

LOW 1_
```

You are now ready to burn 2716 ICs. If you want to burn 2764 ICs you must press [s]. You will see the list of ICs on the screen. Select the 2764.

Prepare the programmer for Intel Hex by typing in [I] [P] [return] (For a Bay Tech prom burner). The Bay Tech will answer back and this information will appear on the Kaypro screen:

```
BAY TECHNICAL ASSOCIATES
MODEL 953B EPROM PROGRAMMER
COPYRIGHT 1982
REV. BMR.2
```

Burn the Prom

Then hold [CTRL] and press [S] to begin the transfer (downloading).

The screen indicates "LOW 1". This tells you that the program is ready for the first transfer to start. After the prom is burned, this prom will go in the lower prom bank.

You can see the hex code on the screen as it is being output to the prom programmer.

When transfer of the object code to the Kaypro serial port is complete, one of two messages will appear:

- More to Transfer

  This tells you that there are more EPROMs to burn for the PAL program.

- Transfer Complete

  This tells you that all the code for the PAL program has been output.

More to Transfer

The screen for More to Transfer is:

```
THERE IS MORE TO TRANSFER

        MENU
C = CONTINUE TO NEXT TRANSFER
R = REPEAT LAST TRANSFER

"RETURN" = RETURN TO MAIN MENU
```

This screen tells you there is more object code to transfer. It gives you three options.

- you can continue with the next block of object code to transfer (type [C])
- you can repeat the last transfer (type [R]) and make a duplicate of the last EPROM that was burned.
- you can go back to the Main Menu, (type [RETURN])

When you and the programmer are ready to burn the next EPROM, press [C]. The menu displayed on the CRT will be:

```
OBJECT OUTPUT READY

KEYBOARD ENTRIES ARE NOW TRANSFERED TO SERIAL PORT.

TYPE WHATEVER IS NEEDED (CONSULT YOUR PROM PROGRAMMERS MANUAL)
TO PROMPT THE PROGRAMMER TO RECEIVE INTEL HEX CODE FROM THE KAYPRO.

HOLD 'CRTL' AND TOUCH 'S' TO START TRANSFER

"ESC" = ABORT AND RETURN TO MAIN MENU

HIGH 1_
```

"HIGH 1," the next block of object code, is ready to transfer. When the programmer has been enabled to receive the Intel Hex format, hold [CTRL] and press [S] to begin the transfer.

The indication of HIGH 1 also means that when the prom is burned, it is plugged in the upper prom bank in the 8400.

Again, if there is more object code to transfer, the message "MORE TO TRANSFER" is given. Continue to output object code until all PROMs have been burned. Label each prom according to its position in the 8400 prom bank. For example, 1-lower, 1-upper, 2-lower, 2-upper, etc. Proms are always plugged into the CRT board, and will be in numeric sequence from left to right.

Transfer Complete

If the transfer of object code is complete, you'll see the following message:

```
TRANSFER COMPLETE

R = REPEAT LAST TRANSFER

"RETURN" = RETURN TO MAIN MENU
```

This gives you two options.

- you can repeat the last transfer of the last block of object code (type [R]) and duplicate the last EPROM that was burned
- you can return to the Main Menu (type [RETURN])

Messages Only Output

Press [M] when on the Hex Transfer Output Menu to select this function.

If you want to make a Messages Only Output, use this function. The serial output will be in Intel Hex format (see table 2 at the end of this section).

Transfer will occur in blocks according to the EPROM size you select in the menus. The menus, screens and messages that you see with Messages Only Output are basically the same as for Object Code Output. The procedures for output are the same.

Source Code Output with Messages

Press [S] when on the Hex Transfer Output Menu to select this function. No compilation is required to transfer out source code.

This function transfers ladder source code, with I/O assignments and messages to the Kaypro serial port. Serial output will have Intel Hex format (see table 2 at the end of this section).

Transfer will occur in blocks according to the EPROM size you select in the menus. This feature was implemented to be compatible with early development equipment. The menus, screens and messages you see in Source Code Output with Messages is basically the same as for Object Code Output. The procedures for output are the same.

PAL.COM System Messages During Output Hex

Occasionally, during output of object code, it is necessary for the Kaypro to do some housekeeping. This may take few seconds. Messages will come up for a short time and then disappear. You may see:

```
SAVING OBJECT TEMPORARILY . . .

RESTORING OBJECT . . .
```

Because of memory constraints in the Kaypro, object code is saved in a disk file during this time. If there isn't enough room on the disk to save the object code, the following message will appear on the CRT screen:

```
SAVING OBJECT TEMPORARILY

DISK FILE FULL, COMMAND ABORTED
```

If you see this message, you'll have to start transfer again. Press any key on the keyboard to return to the Main Menu. Restart on an empty disk that has been initialized or the same error will occur.

4.10.2 Input Hex

Press [I] on the Hex Transfer Input & Output Menu to select the input hex function. The following display will appear on the CRT screen:

```
HEX TRANSFER INPUT

        MENU
    S = SOURCE CODE INPUT (WITH MESSAGES OPTIONAL)
    A = APPEND SOURCE CODE INPUT TO EXISTING LADDER
    M = MESSAGES ONLY INPUT

OR TOUCH "RETURN" TO RETURN TO MAIN MENU
```

Source Code Input (With Messages Optional)

Press [S] when on Hex Transfer Input Menu to select this function. The following menu will appear on the CRT screen:

```
WAITING FOR INPUT...

KEYBOARD ENTRIES ARE NOW TRANSFERED TO SERIAL PORT.

TYPE WHATEVER IS NEEDED (CONSULT YOUR PROM PROGRAMMERS MANUAL)
TO PROMPT THE PROGRAMMER TO TRANSFER THE PROM IN INTEL HEX CODE

EXAMPLE:
"ESC" = ABORT AND RETURN TO MAIN MENU
```

This function makes the Kaypro ready to receive ladder source code with I/O assignments and messages. The device that provides the source code input may not be an EPROM programmer. What is required is that the device give serial input in Intel Hex format and the input must be in blocks of code compatible with EPROM sizes for 8400 controls.

Any ladder source code that is not stored in a disk file, one that is currently in the program, will be destroyed by this command.

The device must be requested to transmit Intel Hex format. Some programmers require commands from the Kaypro serial port to request them to transmit Intel Hex. Other programmers have keyboards which allow requesting Intel Hex format directly.

Each prom programmer has its own way to enable Intel Hex. Refer to your prom programmer manual to determine what is needed here.

Any Kaypro keyboard entry will be transferred to the serial port. This can be used to request the programmer to transmit Intel Hex. The Kaypro will be in terminal mode until transfer begins.

As the ladder source code is received at the serial port, you will see colons on the CRT.

After the ladder source code has been transferred, the following message will appear on the CRT screen:

IS THERE MORE SOURCE CODE TO INPUT? Y/N

If there is more ladder source code to receive, press [Y] on the keyboard. You'll see the following menu:

WAITING FOR INPUT...

KEYBOARD ENTRIES ARE NOW TRANSFERED TO SERIAL PORT.

TYPE WHATEVER IS NEEDED (CONSULT YOUR PROM PROGRAMMERS MANUAL)
TO PROMPT THE PROGRAMMER TO TRANSFER THE PROM IN INTEL HEX CODE

EXAMPLE:
"ESC" = ABORT AND RETURN TO MAIN MENU

Once you've set up the prom programmer, or other device, to output Intel Hex the transfer will begin.

If there is more ladder source code to transfer, press [Y] for yes. If the transfer is complete press [N] for no. If the transfer completed successfully the following message will appear on the CRT screen:

FILE TRANSFER COMPLETED TOUCH 'RETURN' TO RETURN TO MAIN MENU

Possible Error The PAL program will check to see that the source code input contains I/O assignment information. If I/O assignments are not present the following error messages will appear on the CRT:

SOURCE CODE ERROR!
TOUCH 'RETURN' TO RETURN TO MAIN MENU

You can only touch [RETURN] on the Kaypro keyboard to return to the Main Menu page. If there are no I/O assignments in the source code, this error does not occur on Append or Message load.

Append Source Code Input to Existing Ladder

Press [A] when on the Hex Transfer Input menu to select this function.

Any I/O assignments or messages in the source input will be added to the end of any existing ladder already present in the Kaypro. The sequence of operations for this function are similar to Source Code Input.

Messages Only Input

Press [M] when on the Hex Transfer Input Menu to select this function.

Any existing messages already present in the Kaypro will be destroyed by this function. Any I/O assignment or ladder rungs already existing in the Kaypro memory will not be disturbed.

The messages that are input must contain all 40 messages. If all 40 Messages are not present the Message Display and Modify function will not work properly.

As the ladder message input is received at the serial port, colons will be shown on the CRT.

After the ladder message input has been transferred, the following message will appear on the CRT screen:

```
MESSAGE INPUT COMPLETE, NO ERRORS

TOUCH 'RETURN' TO RETURN TO MAIN MENU
```

Possible Error

If the messages are too long, the following will be displayed on the CRT screen:

```
ERROR!!  MESSAGE INPUT TOO LONG

TOUCH [RETURN] TO RETURN TO MAIN MENU
```

If the data input during this time doesn't contain valid message data, the Message Display and Modify function will not work properly.

4.10.3 Terminal

Press [T] when on the Hex Transfer In and Out Menu to select this function. The following message will appear on the CRT screen:

TERMINAL MODE - HOLD "CRTL" AND TOUCH "E" TO RETURN TO MAIN MENU

This function is used to gain access to any additional features that may be provided on a prom programmer.

Any keyboard inputs will be transferred to the serial port and also to the CRT screen. For instance the Bay Technical Associates prom programmer will list a prom from address 0 to 10 by typing

[L] [0] [,] [1] [0] [RETURN].

Refer to your prom programmer manual for these features.

To exit the terminal mode hold [CRTL] and press [E] on the keyboard. This will cause the Main Menu to be displayed on the CRT screen.

4.11 Print

This function is used to print the I/O assignments, PAL program, and messages. Each can be printed separately or all together.

Press [P] when on the Main Menu page to select this function. The following menu will appear on the CRT screen:

SELECT PAGE TO BE PRINTED

I = PRINT I/O ASSIGNMENTS
L = PRINT LADDER
M = PRINT MESSAGES
A = PRINT ALL PAGES (MESSAGES, I/O, LADDER)

"RETURN" = RETURN TO MAIN MENU

- Press [I] to print the I/O assignments.
- Press [M] to print the messages.
- Press [L] to print the PAL program.
- Press [A] to print the I/O assignments, messages, and PAL program all together.

Possible Error

If the printer is not ready (i.e. cable missing or printer is not on line) the following message will appear on the CRT screen:

PRINTER NOT READY

TOUCH 'RETURN' TO RETURN TO MAIN MENU

Press the [RETURN] key on the Kaypro keyboard to return to the Main Menu. Correct the problem with the printer, then try your print out again.

4.12 Using XREFLAD.COM

As mentioned in chapter 4.1, Allen-Bradley provides a file called XREFLAD.COM.

S-BASIC is no longer supplied by Kaypro. It is supplied by Allen-Bradley when you purchase the Kaypro through Allen-Bradley. If you did not purchase the Kaypro through Allen-Bradley, you must purchase S-BASIC separately.

XREFLAD.COM should be part of your Main Disk, as described in chapter 4.1 of this manual.

XREFLAD.COM is used to generate a cross reference between the flags and variables in your PAL program, and the rungs in your PAL program.

Rules

The PAL program that you cross reference must be a PAL source code file stored on your Data Disk.

Only flags and variables that begin with a letter coded are cross referenced to rungs in the PAL program. The cross reference is generated in alphabetical order for the following:

- A001 to A040
- B001 to B080
- C001 to C016
- D001 to D010
- G001 to G040
- H001 to H100
- L001 to L040
- M001 to M040
- N001 to N100
- R001 to R120
- T001 to T016

You can output the cross reference to the Kaypro CRT, or to a printer.

Getting Started

Before you start generating the cross reference, first make sure that any active PAL program (one that is in PAL.COM) is saved to a disk file.

To use XREFLAD.COM, you have to call it up under CP/M. Leaving PAL.COM and going to the CP/M operating system will destroy any PAL program that is active in PAL.COM.

With any active PAL program that you want to keep safely in a disk file, access CP/M. You can do this from the the Main Menu of PAL.COM by holding [CTRL] and typing [C].

Loading XREFLAD

While in CP/M, access XREFLAD.COM by typing

```
XREFLAD [RETURN]
```

Important: This assumes that XREFLAD.COM is on your Main Disk, along with CP/M. Also, the Main Disk is in disk drive A, which is the default drive. We also assume that your Data Disk is in drive B, and the PAL program source code is in a file on this disk.

Enter the Date

XREFLAD should now load, and when loading is complete you should see the following on the CRT:

```
ENTER THE DATE (00/00/00) >? _
```

Type in the date you want to appear with the cross reference. Only the first 8 characters you type will be used. You can see that the program gives you an example before the prompt. When you finish entering the date, press [RETURN].

Enter the PAL Program File Name

Next, the program requests the name of the file that the PAL program source code is in. The CRT shows:

```
ENTER FILE NAME A:FILE.SRC 'RETURN' ? _
```

The program gives you an example before the prompt. Type in the name of the source code file you want to cross reference. Note that you must adhere to CP/M file name conventions; include the disk drive letter and the ":", if appropriate. Type the file name with all caps (the [CAPS LOCK] key should be ON). When you finish entering the file name, press [RETURN].

Enter the Page Heading

The program now requests the message that you want to appear at the top of each page of the cross reference listing. The CRT shows:

```
ENTER THE PAGE HEADING (FILE) >? _
```

Type in a heading. Only the first 47 characters that you enter will be used. When you finish entering the heading, press [RETURN].

Select the Output Device

Next, the program wants to know where you want the listing of the cross reference to occur. You have the choice of listing it on the CRT or on a printer, as the CRT shows:

```
PRINT THE XREF TO CONSOLE (C) OR PRINTER (P) >? _
```

Important: If you choose to list on the CRT (console), the listing will scroll as more data is added. Data that scrolls off the screen cannot be viewed again. To stop the scrolling, hold [CRTL] and press [S]. To resume scrolling, hold [CRTL] and press [S] again. If you select the printer for output make sure the printer is connected and operating properly before selecting the printer option.

- To choose the CRT, type [C] [RETURN].
- To choose the printer, type [P] [RETURN].

Loading the File

When you press [RETURN] after making your choice, the disk drive with the PAL program source file will activate. The file will load, and after it finishes loading the CRT will show:

(Press any key to turn off disk motor if you want.)

The disk drive probably won't be running at this point anyway.

Possible Error

If XREFLAD cannot find your PAL program file on the disk, the following message will appear on the CRT:

S-TYPE FILE NOT FOUND ERROR IN LINE 0108

WARM BOOT

A>

System control has automatically returned to CP/M. All of the previous steps, starting with Loading XREFLAD must be performed again to generate your cross reference. Make sure you have the file name correct, and that the file is on disk.

Waiting for the Output

If the file has been found successfully, the XREFLAD program will start to sort the flags and variables. This process could take up to 10 minutes or longer depending on the length of your PAL program.

Important: There is no visual indication that the sort is in progress.

After the sort is complete, the cross reference listing will be output to the specified output device. When the cross reference output is complete, system control will return to CP/M.

For an Example

For an example of XREFLAD cross reference output, see the last topic in section 3.

5. 8400 PAL Instructions

Each 8400 PAL instruction in this section is defined by its:

- name
- display format
- keyboard entry code
- possible addresses (flags, variables, timers, counters)
- execution time
- number of object code bytes
- description with an example.

If this is your first look at instructions:

- Scan the groups of instructions given below. Try to get an idea of what each group does. If you want to read about each group, look on the page number for the group.

If you are familiar with instructions, but want a refresher:

- Note the page number for the group of the instruction you want.
- Look in the group for the instruction you want.

Instructions are grouped into these areas:

5.1 Examine Instructions

Examine instructions are always:

- part of an input group on a rung
- used to test the state of flags or timers, or real I/O.

They form true preconditions (continuous logic paths) when their addressed flags or timers are:

- internal logic 1 = examine on (-] [-) true precondition
- internal logic 0 = examine off (-]/[-) true precondition

PAL operates on a "snapshot" of the inputs. Real inputs (R flags and V variables) are read into PAL at the beginning of each ladder pass. The PAL program is performed, then the real outputs are updated at the end of the ladder pass.

Pay special attention to how the examine instructions work with inverted I/O and real inputs and outputs. We recommend you use normal I/O.

Flag or Timer
-] [-

Examine On

Keyboard Entry Code.....01
Possible Flags or Timers...R001-R120, B001-B080, T001-T016
M001-M040, N001-N100, H001-H100
Execution Time (usec.).....3.0
Object Code Bytes.....4

An examine on instruction forms a true precondition (a continuous logic path) when its addressed flag or timer is true (=1 and "on" in PAL with normal I/O). It forms a false precondition (a broken logic path) when its addressed flag or timer is false (=0 and "off" in PAL with non-inverted I/O).

In this rung, if flag B008 is true (=1), the examine on instruction forms a true precondition. The rung has a continuous logic path and the OTE (output energize) instruction sets real flag R100 to 1 until B008 again becomes false (= on in PAL if you have selected normal I/O).

Remember, if you select inverted I/O, real inputs and outputs are "true" and "on" when they are equal to a logic 0 (false and off in PAL). They are "false" and "off" when they are equal to a logic 1 (true and on in PAL).

If you select inverted I/O and a real input R flag is 0 (on for inverted I/O), an examine on instruction monitoring that flag forms a false precondition. We recommend that you use normal I/O, whenever possible, to avoid any confusion.

Flag or Timer
-]/[-

Examine Off

Keyboard Entry Code.....02
Possible Flags or Timers...R001-R120, B001-B080, T001-T016
M001-M040, N001-N100, H001-H100
Execution Time (usec.).....4.0
Object Code Bytes.....8

An examine off instruction forms a true precondition (a continuous logic path) when its addressed flag or timer is false (=0 and off in PAL with normal I/O). It forms a false precondition (a broken logic path) when its addressed flag or timer is true (=1 and on in PAL with normal I/O).

In this rung, if flag B009 is false (=0), the examine off instruction forms a true precondition. The rung has a continuous logic path and the OTD (output de-energize) instruction resets the real flag R100 to 0 until B009 again becomes true.

Remember, if you select inverted I/O, real inputs and outputs are "true" and "on" when they are equal to a logic 0 (false and off in PAL). They are "false" and "off" when they are equal to a logic 1 (true and on in PAL).

If you have selected inverted I/O and a real input R flag is 1 (off for inverted I/O), an examine off instruction monitoring that flag forms a false precondition. We recommend that you use normal I/O, whenever possible, to avoid any confusion.

5.2 Transfer Execution Instructions

These instructions allow you to jump forward in PAL program execution. They can be used to skip portions of the ladder that only need to be performed at certain times.

Backward jumps are not allowed. A LBL instruction must follow its GTO in the current ladder pass.

Important: the value of L-words, A-words, B-words, N-words, and R-words, that are only set and reset in a GO-TO loop will not change when the loop is jumped over.

GO-TOs are important because they reduce the execution time of the PAL program.

Destination Label Number
-(GTO)-

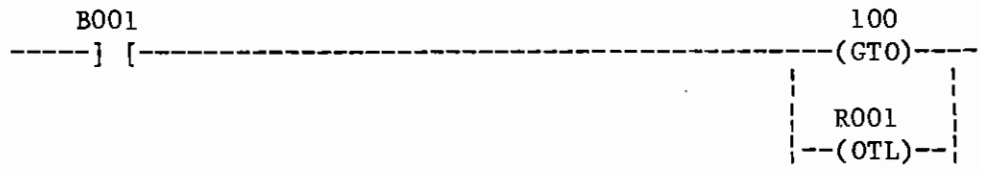
GTO (Go To Label)

Keyboard Entry Code.....89
Possible Immediate Values..001-999
Execution Time (usec.).....4.6
Object Code Bytes.....7

When rung preconditions are true, the GTO instruction transfers ladder execution to a rung with a corresponding LBL instruction. The value of the GTO instruction must match the value of the desired label. The immediate value of the GTO and the immediate value of the LBL must be the same. The LBL instruction must follow the GTO in the current ladder pass.

GTO and LBL are used to skip portions of the ladder that don't need to be executed unless certain conditions are present. When preconditions of the rung are false, the GTO has no effect and ladder execution resumes with the next rung in the ladder.

If the GTO instruction is one of several parallel outputs on a rung, it should be the last sub-rung in the output group. If the GTO is not the last sub-rung, the sub-rungs that follow the GTO will not be executed when rung preconditions are true.



In this rung, when rung preconditions are true the GTO is executed first and the OTL (output latch) of R001 is never performed.

For an example of using GTO and LBL, see the definition of the LBL instruction on the next page.

Immediate value
-(LBL)-

LBL (Label)
Keyboard Entry Code.....80
Possible Immediate Values..001-999
Execution Time (usec.).....0.0
Object Code Bytes.....0

A LBL instruction is used as an address, or locater, to allow transfers in ladder rung execution. It is used exclusively with the GTO instruction. The GTO instruction can only reference a LBL instruction that occurs after it is in the current ladder pass.

```
   FLTCLR    ESTOPSW
     H044      R072                                          100
----]/[------]/[-----------------------------------------(GTO)----
                                    .
                                    .

                                Reset Rungs
                                    .
                                    .

     100                                                    B001
---(LBL)-------------------------------------------------(OTE)----
```

In this example, the LBL and the GTO instructions skip portions of ladder that only require execution at certain times. If flag H044 (fault clear from the CNC) is false (=0), and input R072 (the Emergency Stop switch) is false (=0), the GTO instruction transfers execution of the ladder to the 100 LBL instruction. GTO and LBL skip the section of ladder that performs reset functions.

The LBL instruction can be in a rung of its own, controlling an output instruction that addresses a flag (as in the example above), or it can have other instructions following it that perform any standard ladder function. The LBL instruction cannot be placed in parallel with other instructions, and should be the first instruction in a rung. If it is not the first instruction in the rung, the instructions preceding it in the rung will be skipped.

```
    B001      100                                           B002
----] [-----(LBL)-------------------------------------------(OTE)----
```

In this rung, the examine on of B001 is not performed when the GTO 100 instruction is performed earlier in the ladder pass.

5.3 Control Instructions

Control instructions are always:

- part of an output group on a rung
- used to control flags.

They affect flags differently depending on whether rung preconditions are true or false.

- OTE and OTD toggle flags according to the rung preconditions
- OTL and OTU operate only when rung preconditions are true

During one ladder pass, the last OTE (or OTD) wins. If more than one rung has an OTE of a flag, only the last one has any real effect on the flag.

PAL operates on a "snapshot" of the inputs. Real inputs (R flags and V variables) are read into PAL at the beginning of each ladder pass. The PAL program is performed, then the real outputs are updated at the end of the ladder pass.

Pay special attention to how the control instructions work with inverted I/O and real inputs and outputs. We recommend you use normal I/O.

```
    Flag
   -(OTE)-
```

OTE (Output Energize)

Keyboard Entry Code.....10
Possible Flags.....R001-R120, B001-B080, M001-M040, N001-N100
Execution Time (usec.).....2.0
Object Code Bytes.....3

When rung preconditions are true, the OTE instruction sets its addressed flag to true (=1). If rung preconditions are false, flag is automatically reset false (=0).

Internally, flags are updated for the preconditions of the rung each time the rung is executed. PAL can examine the change in state of these flags during the current ladder pass.

```
     FEEDHOLD                                          FHLD2
     B040                                               N025
-----] [-----------------------------------------------(OTE)---
```

In this rung, if B040 is true (=1), N025 will be energized (made true = 1). If B040 goes false after this rung, N025 will remain energized (true = 1) until the next time this rung is executed and then it will be de-energized (false = 0). See example 1 in the timing diagrams on the next page.

For R flags used with OTE, the state of the real output, corresponding to the flag, is updated at the end of the ladder pass in which the rung preconditions change. So, if the rung preconditions for an OTE of an R flag change, PAL and the CNC will register the change, but the real output won't be affected until immediately after the current ladder pass has ended. PAL can examine the internal change during the current ladder pass.

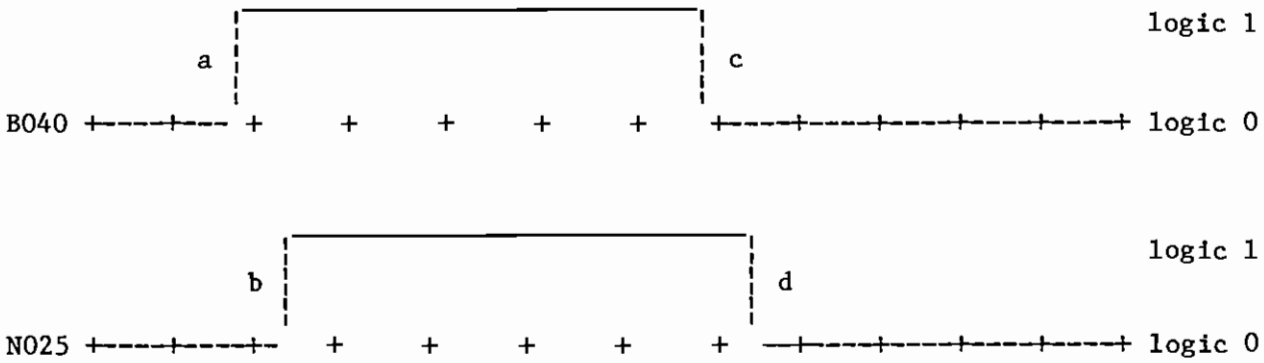
```
     COOLANT                                        COOLANT ON
      B050                                              R080
-----] [-----------------------------------------------(OTE)---
```

In this case, if B050 is true (=1), the real output R080 is energized (=1) at the end of the current ladder pass (when all real outputs are updated). See example 2 in the timing diagrams on the next page.

TIMING DIAGRAMS

EXAMPLE 1: INTERNAL FLAGS (B, M, and N) WITH OTE

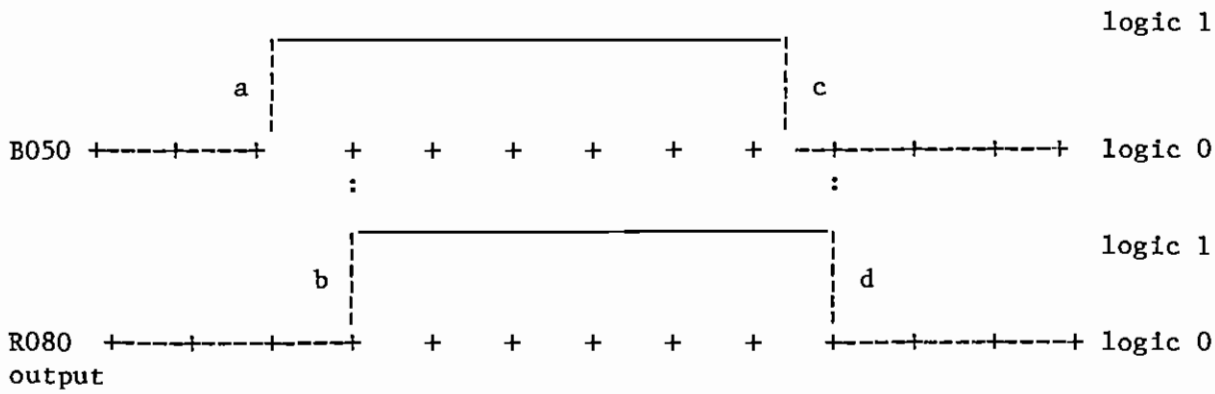
NOTE: + = one ladder pass = 40 ms.



a) B040 set true (sometime after execution of the rung)
b) N025 set true next time rung is executed
c) B040 reset false
d) N025 reset false next time rung is executed

EXAMPLE 2: REAL OUTPUTS (R) WITH OTE

NOTE: + = one ladder pass = 40 ms.



a) B050 set true
b) R080 real output set true at end of current ladder pass
c) B050 reset false
d) R080 real output reset false at end of current ladder pass

```
    Flag
   -(OTD)-
```

OTD (Output Deenergize)

Keyboard Entry Code.....11
Possible Flags.....R001-R120, B001-B080
M001-M040, N001-N100
Execution Time (usec.).....4.6
Object Code Bytes.....7

When the rung preconditions are true, the OTD instruction resets its addressed flag to false (=0) . If rung preconditions are false, the addressed flag is automatically set true (=1).

Internally, flags are updated for the preconditions of the rung each time the rung is executed. PAL can examine the change in state of these flags during the current ladder pass.

```
GUARD DOOR OPEN                                  "GUARD OPEN"
     R093                                            M20
---------]/[-----------------------------------------(OTD)----
```

In this rung, if R093 is false (=0), the examine off instruction will form a true precondition. Message flag M020 is false because of the OTD instruction. If R093 goes true (=1), the examine off instruction will form a false precondition. M020 will energize because the preconditions for the OTD are false. See example 1 in the timing diagrams on the next page.

For R flags used with OTD, the state of the real output, corresponding to the flag, is updated at the end of the ladder pass in which the rung preconditions change. So, if the rung preconditions for an OTD of an R flag change, PAL and the CNC will register the result, but the real output won't be affected until immediately after the current ladder pass has ended.

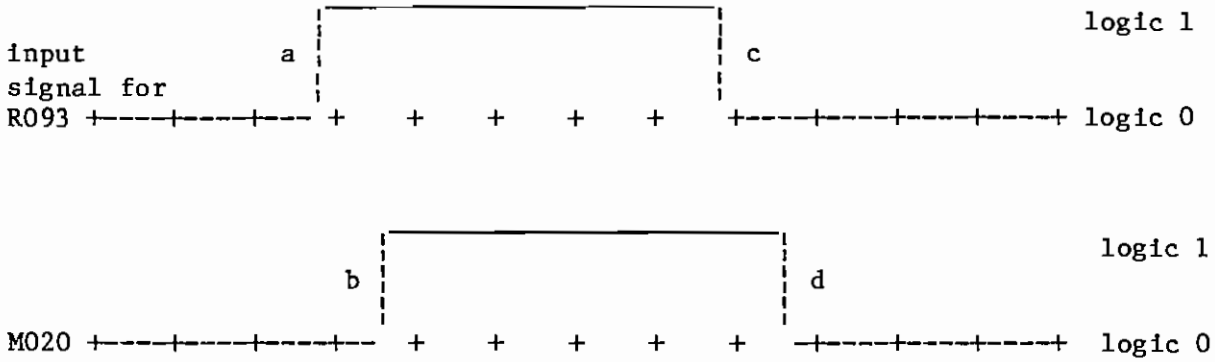
```
"GUARD OPEN"                                     WARNING LIGHT
    M020                                             R070
--------]/[-------------------------------------------(OTD)-----
```

In this case, as long as M020 is false (=0) the examine off instruction forms a true precondition. R070 will be de-energized (=0). When M020 goes true (=1), the real output for R070 will be energized (=1) at the end of the ladder pass. See example 2 in the timing diagrams on the next page.

TIMING DIAGRAMS

EXAMPLE 1: INTERNAL FLAGS (B, M, and N) WITH OTD

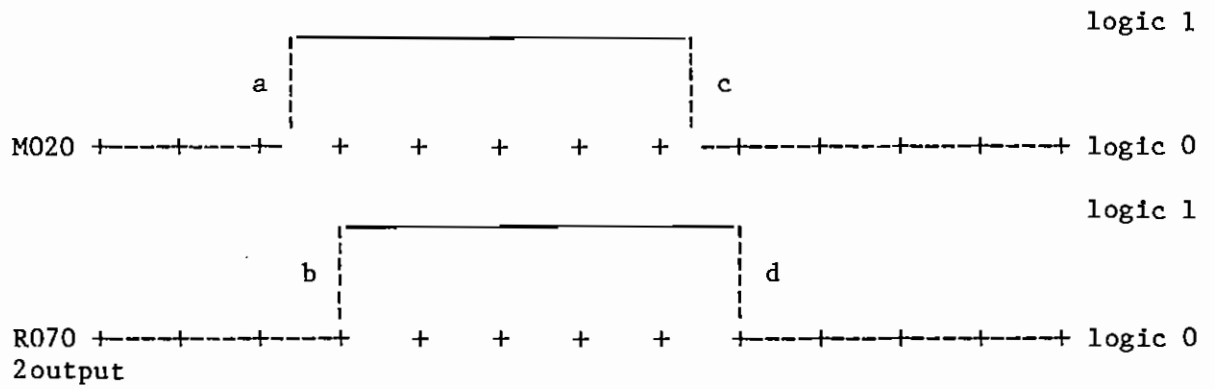
NOTE: + = one ladder pass = 40 ms.



a) R093 input signal goes true (R093 flag updated at the beginning of next ladder pass.
b) M020 set true when the rung is executed.
c) R093 input signal goes false (R093 flag updated at the beginning of next ladder pass.
d) M020 made false when the rung is executed.

EXAMPLE 2: REAL OUTPUTS (R) WITH OTD

NOTE: + = one ladder pass = 40 ms.



a) M020 goes true
b) R070 real output set true at the end of the current ladder pass
c) M020 goes false
d) R070 real output reset false at the end of the current ladder pass

```
    Flag
    -(OTL)-
```

OTL (Output Latch)

Keyboard Entry Code.....12
Possible Flags.....R001-R120, B001-B080
M001-M040, N001-N100
Execution Time (usec.).....4.6
Object Code Bytes.....7

When rung preconditions are true, the OTL instruction sets its addressed flag to true (=1). A flag that is set true with the OTL instruction will remain true until it is reset false (=0) by an OTU (output unlatch) instruction or power is cycled off.

When the rung preconditions are false, the OTL instruction has no affect on the flag. If the flag is true, it remains true. If it is false, it remains false.

```
    DR.ACK                                                PWACK
    R050                                                  N037
----] [-----------------------------------------------(OTL)---
```

In this rung, if the real input R050 (that monitors servo drive power) is true (=1), N037 (power acknowledge to the CNC) is latched true (=1). See example 1 in the timing diagrams on the next page.

```
    SPEN      SPIN.INTERLOCK                  SPINDLE ENABLE
    H038          B020                            R090
-----] [----------]/[---------------------------(OTL)-------
```

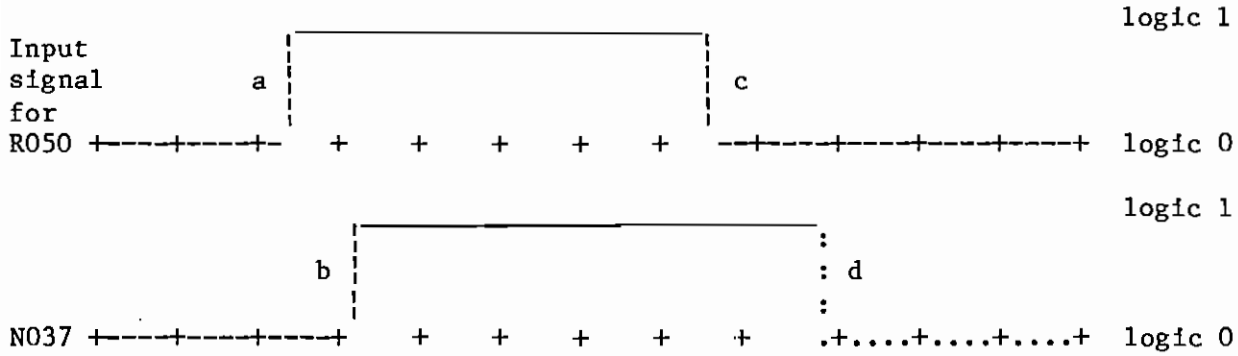
In this rung, if the CNC to PAL flag for spindle enable, H038 is true (=1), and if the internal flag for spindle interlock, B020, is false (=0), rung preconditions are true. R090 will be latched true (=1) by the OTL instruction. If the rung preconditions change, R090 is not affected. The R090 real output would have to be unlatched with a different rung elswhere in PAL. See example 2 in the timing diagrams on the next page.

Remember that the state of real outputs is updated immediately after the current ladder pass. If R090 is latched true in one pass, any change is not registered in the corresponding real output until the end of the pass.

TIMING DIAGRAMS

EXAMPLE 1: INTERNAL FLAGS (B, M, and N) WITH OTL

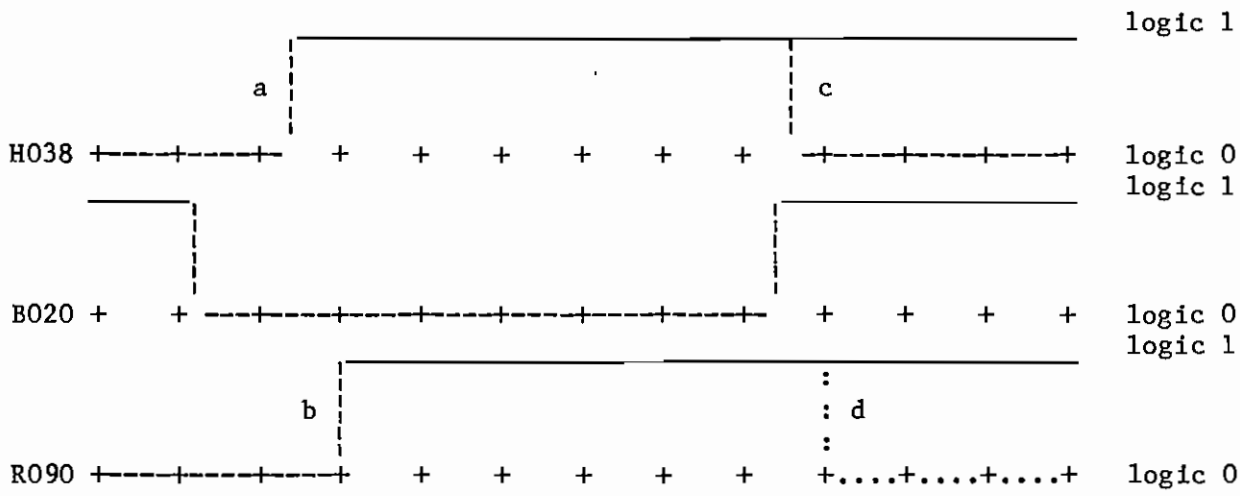
NOTE: + = one ladder pass = 40 ms.



a) Input signal for R050 goes true. The real input corresponding to R050 is read by PAL at the beginning of the next ladder pass.
b) N037 is latched true the next time the rung is executed.
c) Input signal for R050 goes false. The real input corresponding to R050 is read by PAL at the beginning of the next ladder pass
d) N037 is not affected by the rung (although it would probably be unlatched shortly after by another ladder rung).

EXAMPLE 2: REAL OUTPUTS (R) WITH OTL

NOTE: + = one ladder pass = 40 ms.



a) H038 goes true
b) With B020 false, R090 real output is latched true when executed and outputted at the end of the current ladder pass.
c) B020 goes true and H038 goes false
d) R090 real output is not affected by the change in rung preconditions (although R090 would probably be unlatched by another ladder rung shortly after in PAL).

```
        Flag
        -(OTU)-
```

OTU (Output Unlatch)

Keyboard Entry Code.....13
Possible Flags.....R001-R120, B001-B080
M001-M040, N001-N100
Execution Time (usec.).....4.8
Object Code Bytes.....9

When rung preconditions are true, the OTU instruction resets the state of its addressed flag to false (=0).

When rung preconditions are false, the OTU instruction has no effect on the flag.

```
     DRACK                                              PWACK
      R050                                              N037
-----]/[--------------------------------------------(OTU)----
```

In this rung, if the real input corresponding to flag R050 is false (=0), the examine off instruction forms a true precondition. The PAL to CNC power acknowledge flag, N037 (PWAK), is unlatched (reset false (=0)). See example 1 in the timing diagrams on the next page.

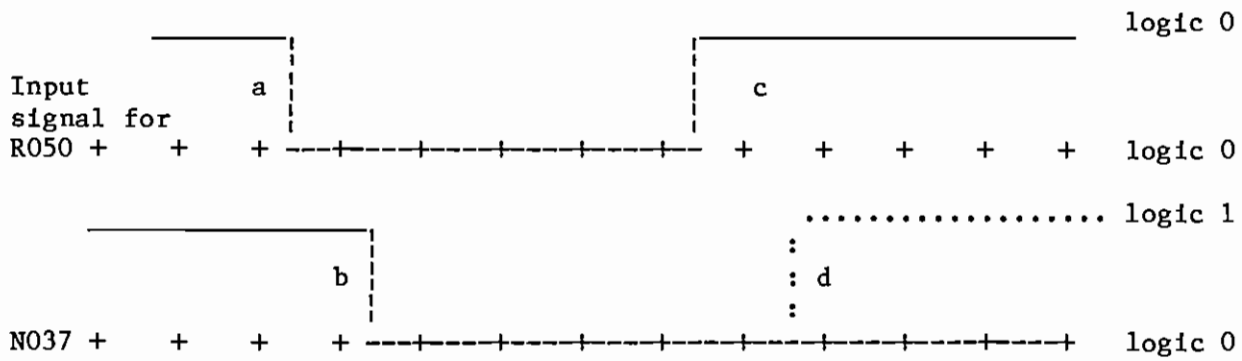
```
     SPEN                                       SPINDLE ENABLE
     H038                                            R090
----]/[-----------------------------------------(OTU)------
```

In this example, if the CNC to PAL flag for spindle enable, H038 (SPEN) is false (=0), R090 will be unlatched. The real output R090 is updated at the end of the current ladder pass. If H038 goes true (=1) sometime later, it will have no affect on R090 in the rung shown above. R090 would have to be latched true (=1) by another rung. See example 2 in the timing diagrams on the next page.

TIMING DIAGRAMS

EXAMPLE 1: INTERNAL FLAGS (B, M, and N) WITH OTU

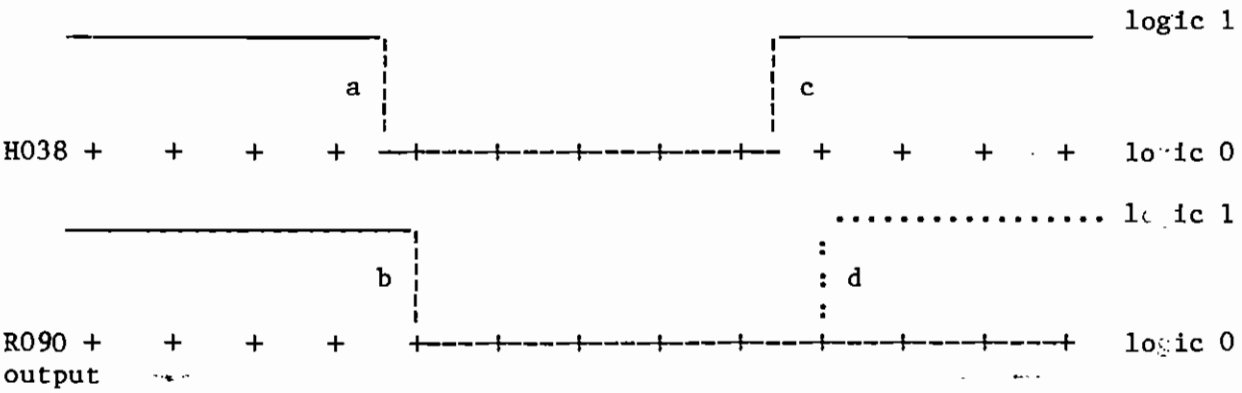
NOTE: + = one ladder pass = 40 ms.



a) Input signal for R050 goes false. Update of real input flag R050 occurs at the beginning of the next ladder pass.
b) N037 is unlatched the next time the rung is executed (after the R050 flag has been updated).
c) Input signal for R050 goes true. Update of real input R050 occurs at the beginning of the next ladder pass.
d) N037 is not affected by the change in rung preconditions (although N037 would probably be latched by another ladder rung shortly after).

EXAMPLE 2: REAL OUTPUTS (R) WITH OTU

NOTE: + = one ladder pass = 40 ms.



a) H038 goes false
b) R090 real output is unlatched when executed and outputted at the end of the current ladder pass.
c) H030 goes true
d) R090 remains unlatched. The change in rung preconditions has no effect on R090 (although it would probably be latched in another rung shortly after).

5.4 GET and PUT Instructions

GET and PUT instruction are used to move values around in PAL.

GET instructions are always part of an input group in a rung. They copy the value of an addressed variable or counter, then load the accumulator with it.

PUT instructions are always part of an output group in a rung. They take values from the accumulator and place them in the variables of counters.

Values in PAL are:

- 16 bit unsigned binary single length words
- 32 bit unsigned binary double length words
- 16 bit BCD values having a 4 digit decimal equivalent
- decimal immediate values from 0 to 65535

Values are found in:

- the accumulator, a 16 bit or 32 bit internal register used to hold values during operations in PAL
- variables, 16 bit and 32 bit registers allocated in PAL for the storage of values
- counters, 16 bit binary up/down false to true precondition edge triggered counters

Most values in PAL correspond to 16 bit unsigned binary words and have a range of 0 to 65535 decimal. Double length values, 32 bit unsigned binary words, are generated from multiply or divide instructions, and can be used. When they are used, they are put in D variables for internal storage, and moved with double length get and double length put instructions. Double length values are useful with the multiply instruction when calculation results in values greater than 65535 decimal.

Variable or Counter
-(GET)-

GET Keyboard Entry Code.....40
Possible Variables or Counters.....A001-A040,
C001-C016, L001-L040
G001-G040, V001-V040
Execution Time (usec.).....2.8
Object Code Bytes.....4

A GET instruction copies the value of its addressed variable or counter and writes it into the accumulator.

```
    MFUNC                                              COOLANT
    G001      00008                                      B008
---(GET)------(EQI)-----------------------------------(OTE)---
```

In this rung, the GET instruction copies the current value of G001 and writes it into the accumulator. The EQI (equal immediate) instruction compares the contents of the accumulator with 8. If the accumulator contains 8, the EQI instruction forms a true precondition. The rung has a continuous logic path and the OTE (output enable) instruction sets flag B008 true (=1). If G001 has any value other than 8, flag B008 is reset false (=0).

This rung would normally only be executed if an H033 (MSTR) flag was true (=1) from the CNC.

Immediate value
-(GTI)-

GTI (Get Immediate)

Keyboard Entry Code.....50
Possible Immediate Values..00000-65535
Execution Time (usec.).....0.8
Object Code Bytes.....3

A GTI instruction writes its immediate value into the accumulator.

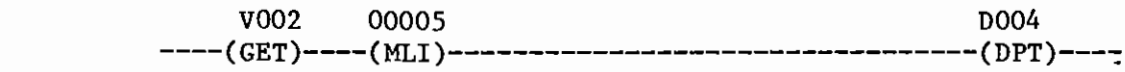
In this rung, the GTI instruction writes 2000 into the accumulator. The PUT instruction takes the value in the accumulator and puts it into variable L002. The L002 variable is a PAL to CNC word for the requested spindle RPM.

DOUBLE VARIABLE
-(DGT)-

DGT (Double Length Get)

Keyboard Entry Code.....60
Possible Double Variables..D001-D010
Execution Time (usec.).....6.4
Object Code Bytes.....10

A DGT instruction copies the value of its addressed variable and writes it into the accumulator.



In this rung, the GET instruction writes the variable V002 into the accumulator. The MLI instruction multiplies its immediate value by the value in the accumulator. The DPT instruction puts the product into double variable D004.

A DGT instruction is similar to a GET, except that a DGT copies a 32-bit word and a GET copies a 16-bit word. For 32-bit operations, you must use a DGT with DPT (double length put). If you use DGT with a PUT (single length put), only the lower 16 bits are transferred into the variable that PUT addresses.

Variable or Counter
-(PUT)-

PUT Keyboard Entry Code.....49
Possible Variables or Counters.....C001-C016,
L001-L040, A001-A040
Execution Time (usec.).....4.6
Object Code Bytes.....8

When rung preconditions are true, the PUT instruction puts the current value of the accumulator into its addressed variable or counter.

If rung preconditions are false, the PUT instruction will be ignored.

```
     TSTR2      TLN02                                  TOOL NO.
     H037       G005                                    V005
----] [-------(GET)-----------------------------------(PUT)----
```

In this rung, if a T-code is currently active, H036 (T-strobe from the CNC) will be true (=1). The tool number that is currently programmed (G005) is written into the accumulator.

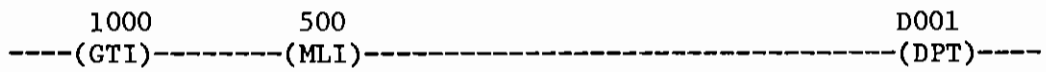
Double variable
-(DPT)-

DPT (Double Length Put)

Keyboard Entry Code.....69
Possible Double Variables..D001-D010
Execution Time (usec.).....8.4
Object Code Bytes.....14

When rung preconditions are true, the DPT instruction puts the current value of the accumulator into the addressed variable. If rung preconditions are false, the DPT instruction will be ignored. The DPT instruction is used with 32 bit operations.

The DPT instruction operates like a PUT, except the word length is doubled to 32 bits. The DPT instruction can be used with the DGT (double length get) instruction and still maintain 32 bit operations. Also, the DPT instruction must be used with the MLI (multiply) instruction when the result may exceed the value of 65535.



This rung loads 1000 into the accumulator, multiplies it by the immediate value 500. This number is more than 16 bits long, (single word length) so you must use the DPT instruction to put this number into double length variable D001 and still store the complete number in the 32 bit register.

5.5 Compare Instruction

Compare instructions are always part of an input group in a rung.

They address single length variables, counters or immediate values and form true preconditions when:

- the value is equal to the accumulator, EQL and EQI
- the value is less than the accumulator, LES and LSI.

Variable or Counter
-(EQL)-

EQL (Equal)

Keyboard Entry Code.....45
Possible Variables or Counter.....A001-A040, C001-C016, L001-L040, G001-G040
Execution Time (usec.).....5.2
Object Code Bytes.....8

The EQL instruction compares the value of its addressed variable or counter with the value in the accumulator. It forms a true precondition if they are equal. If the values are not equal, the EQL instruction forms a false precondition.

```
     S1REQ     S1RPM                              SP. SPEED OK
     L002      G007                                    B010
----(GET)----(EQL)-----------------------------------(OTE)----
```

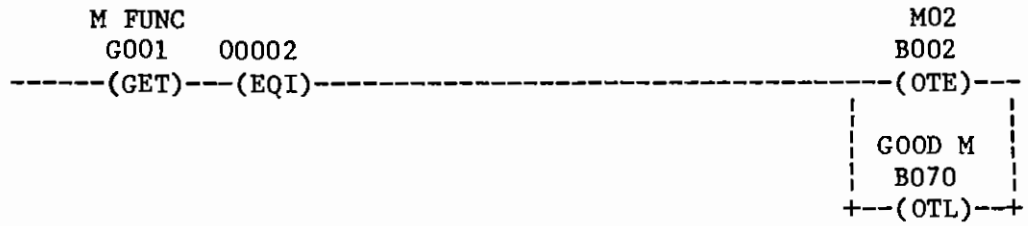
In this rung, the GET instruction writes the value of L002 (spindle speed demand from PAL) into the accumulator. The EQL instruction compares the value in the accumulator (L002) with the value in G007 (actual spindle speed from the CNC). If the two values are equal, rung preconditions are true and the OTE instruction sets flag B010 true (=1). If the values are not equal, rung preconditions are false and B010 will automatically be reset false (=0).

Immediate value
-(EQI)-

EQI (Equal Immediate)

Keyboard Entry Code.....55
Possible Immediate Values..00000-65535
Execution Time (usec.).....4.0
Object Code Bytes.....8

The EQI instruction compares its immediate value with the value in the accumulator. If they are equal, the EQI instruction forms a true precondition.



This ladder sequence might be part of an M-function decoding section. The GET instruction writes the value of G001 (M-function number from the CNC) into the accumulator. The EQI instruction compares its immediate value (2) with the value in the accumulator (G001). If the values are equal, rung preconditions are true, and flag B002 will be set true (=1). PAL flag B070 is latched true (=1) at the same time.

If the current M-function number is not equal to 2, rung preconditions are false and B002 is automatically reset false (=0). Flag B070 is not automatically reset when rung preconditions are false. It must be unlatched false elsewhere in the ladder program, perhaps to indicate an invalid M-code.

Variable or Counter
-(LES)-

LES (Less Than)

Keyboard Entry Code.....46
Possible Variables or Counters.....A001-A040, C001-C016, L001-L040, G001-G040
Execution Time (usec.).....5.2
Object Code Bytes.....8

The LES instruction compares the value in the accumulator with the value of its addressed variable. If the accumulator is less than the variable, the LES instruction forms a true precondition. If the accumulator is equal to or greater than the variable, the LES instruction forms a false preconditon.

```
     S1RPM     SPMIN                          SPINDLE TOO SLOW
     G007      L007                                M015
---(GET)----(LES)-------------------------------(OTE)-------
```

In this rung, the GET instruction writes the value of G007 (the actual spindle speed based on encoder feedback) into the accumulator. The LES instruction compares L007 (the minimum spindle RPM for the current gear range) with the accumulator (G007). If the accumulator is less than L007, message flag M015 is set true (=1), and the message will be shown on the CRT.

If the accumulator is greater than L007, the message flag is automatically reset false (=0). The message is not shown because actual spindle speed is above the minimum allowed value.

Immediate value
-(LSI)-

LSI (Less Than Immediate)

Keyboard Entry Code.....56
Possible Immediate Values..00000-65535
Execution Time (usec.).....4.0
Object Code Bytes.....8

The LSI instruction compares the accumulator with its immediate value. If the accumulator is less than the immediate value, the LSI instruction forms a true precondition. If the accumulator is equal to or greater than the immediate value, the LSI instruction forms a false precondition.

```
    M FUNC
    G001   00080                                    048
---(GET)---(LSI)-----------------------------------(GTO)---
```

In this rung, the GET instruction writes the value of G001 (the current M-code programmed) into the accumulator. The LSI instruction compares its immediate value, 80, with the accumulator (G001). If the accumulator is less than 80, rung preconditions are true and ladder execution is transferred to a rung with a LBL - 048 instruction by the GTO. If the accumulator is equal to or greater than 80, ladder execution continues with the next rung.

(A)

```
    M FUNC
    G001   00080                                    048
---(GET)---(LSI)-----------------------------------(GTO)---
```

(B)

```
    M FUNC
    0079   G001                                     048
---(GTI)---(LES)-----------------------------------(GTO)---
```

(A)= GTO 048 FOR G001 < 80

(B)= GTO 048 FOR G001 $\geq$ 80

5.6 Math Instructions

Math instructions are always part of an input group in a rung. They add, subtract, multiply, and divide values in PAL.

Math instructions address variables, counters or immediate values and perform their operations with the addressed value and the value in the accumulator. When their operation is finished, they load the accumulator with the result.

The values used by math instructions are usually 16 bit unsigned binary values, but multiply and divide instructions use 32 bit values (double length words). With multiply and divide instructions, the accumulator automatically expands to handle the extra bits.

You should always know in advance the type and range of values your math operations will deal with. Mixing 32 and 16 bit operations could result in a "loss" of bits, perhaps, by trying to put a 32 bit word into a 16 bit variable. In this case, only the lower 16 bits of the 32 bit word will transfer.

Math operations in PAL are <u>not</u> signed. There is no overflow indication. For example, with 16 bit math:

65535 + 1 = 0

0 - 1 = 65535

If operations could result in a negative number, expect to have to check a positive number later on.

PAL does integer math only. Any fractional result is truncated to the whole number part. For example,

23/5 x 10 = <u>40</u> -- <u>truncated</u> in PAL at 23/5 = 4

23 x 10/5 = 46 -- in PAL, which is the correct result

You can see that the order of your math instructions becomes important. In the example above, the multiply instruction had to come before the divide to avoid a truncated result.

Multiply and divide instructions require a relatively large amount of execution time. They should be used sparingly and only when needed. Recommended practice for math instructions:

- Read G, V, and L variables
- Put the values into A variables
- Use the A variables for your math
- After math, put the result where it is needed

```
Variable or Counter
    -(ADD)-
```

ADD

Keyboard Entry Code.....41
Possible Variables or Counters.....A001-A040,
C001-C016, L001-L040,
G001-G040
Execution Time (usec.).....3.0
Object Code Bytes.....4

An ADD instruction adds the value of its addressed variable or counter to the current value in the accumulator, then writes the sum back into the accumulator.

```
     A006    A007                                        L005
----(GET)---(ADD)-----------------------------------------(PUT)----
```

In this rung, the GET instruction copies the value of A006 and stores it in the accumulator. The ADD instruction adds the value of A007 to the contents of the accumulator (A006), and writes the sum back into the accumulator. The PUT instruction copies the value in the accumulator (A006 + A007) and puts it into variable L005. Each time this rung is executed, the value of L005 equals the sum of A006 and A007.

Immediate Value
-(ADI)-

ADI
(Add Immediate)

Keyboard Entry Code.....41
Possible Immediate Values..00000-65535
Execution Time (usec.).....0.8
Object Code Bytes.....4

An ADI instruction adds its immediate value to the value in the accumulator, then writes the sum back into the accumulator.

```
     A030      00512                                    A017
----(GET)----(ADI)-------------------------------------(PUT)----
```

In this rung, the GET instruction writes the value of A030 into the accumulator. The ADI adds its immediate value (512) to the value in the accumulator (A030), then writes the sum back into the accumulator. The PUT instruction puts the value of the accumulator (A030 + 512) into variable A017. Each time this rung is executed, the value of A017 equals the sum of A030 and 512.

Variable or Counter
-(SUB)-

SUB (Subtract)

Keyboard Entry Code.....42
Possible Variables or Counters.....A001-A040, C001-C016, L001-L040, G001-G040
Execution Time (usec.).....3.0
Object Code Bytes.....4

A SUB instruction subtracts the value of its addressed variable or counter from the value in the accumulator, then writes the difference back into the accumulator.

```
     00015    A004                                        L001
----(GTI)----(SUB)-----------------------------------(PUT)----
```

In this rung, the GTI instruction writes its immediate value (15) into the accumulator. The SUB instruction subtracts the value of its addressed variable from the accumulator (15), then writes the difference back into the accumulator. The PUT instruction puts the value of the accumulator (15 - A004) into variable L001. Each time this rung is executed, the contents of L001 equal the difference between 15 and the value of A004.

Immediate Value
-(SBI)-

SBI (Subtract Immediate)

Keyboard Entry Code.....52
Possible Immediate Values..00000-65535
Execution Time (usec.).....0.8
Object Code Bytes.....4

An SBI instruction subtracts its immediate value from the value in the accumulator, then writes the differences back into the accumulator.

```
     A028     00147                                    V025
----(GET)----(SBI)-----------------------------------(PUT)----
```

In this rung, the GET instruction writes the value of A028 into the accumulator. The SBI instruction subtracts its immediate value (147) from the value in the accumulator (A028), then writes the result back into the accumulator. The PUT instruction puts the value in the accumulator (A028 - 147) into variable V025. Each time this rung is executed, the value of V025 equal the difference between A028 and 147.

Variable or Counter
-(MUL)-

MUL (Multiply)

Keyboard Entry Code.....43
Possible Variables or Counters.....A001-A040, C001-C016, L001-L040, G001-G040
Execution Time (usec.).....31.0
Object Code Bytes.....10

A MUL instruction multiplies the value of its addressed variable by the value in the accumulator, and writes the product back into the accumulator.

```
     A004      A011                                        D003
----(GET)----(MUL)-------------------------------------(DPT)----
```

In this rung, the GET instruction copies the value of A004 and writes it into the accumulator. The MUL instruction multiplies the value of A011 by the contents of the accumulator (A004), then writes the result back into the accumulator. The DPT (double length put) instruction writes the contents of the accumulator (A004 x A011) into double length variable D003. Each time this rung is executed, D003 contains the product of the A004 and A011.

The DPT instruction is used because the product of A004 and A011 might be greater than 65535, the maximum number in 16 bit operations. If the PUT instruction were used, only the lower 16 bits of the product would be stored.

Immediate Value
-(MLI)-

MLI (Multiply Immediate)

Keyboard Entry Code.....53
Possible Immediate Values..00001-65535
Execution Time (usec.).....30.6
Object Code Bytes.....11

A MLI instruction multiplies its immediate value by the value in the accumulator, then writes the product back into the accumulator.

```
     D003     29165                                    D002
----(DGT)----(MLI)-------------------------------------(DPT)----
```

In this rung, the DGT instruction copies the value of D003 and writes it into the accumulator. The MLI instruction multiplies its immediate value (29165) by the value in the accumulator (D003), then writes the result back into the accumulator. The DPT instruction writes the value in the accumulator (D003 x 29165) into variable D002. Each time this rung is executed, D002 contains the product of D003 and 29165.

Variable or Counter
-(DIV)-

DIV (Divide)
Keyboard Entry Code.....44
Possible Variables or Counters.....A001-A040, C001-C016, L001-L040, G001-G040
Execution Time (usec.).....40.0
Object Code Bytes.....17

A DIV instruction divides the value in the accumulator by the value of its addressed variable or counter, then writes the quotient back into the accumulator.

```
     L004     G002     A003                                  L002
-----(GET)---(MUL)---(DIV)-------------------------------(PUT)----
```

In this rung, the GET instruction writes the value of L004 into the accumulator. The MUL instruction multiplies the value of G002 by the value in the accumulator (L004) and writes the product back into the accumulator. The DIV instruction divides the value in the accumulator (L004 x G002) by the value of A003, then writes the quotient back into the accumulator. The PUT instruction puts the value in the accumulator ((L004 x G002)/A003) into variable L002. Each time this rung is executed, the value of L004 is multiplied by the value of G002, this product is divided by the value of A003, then the quotient is written into variable L002.

Immediate Value
-(DVI)-

DVI (Divide Immediate)

Keyboard Entry Code.....54
Possible Immediate Values..00001-65535
Execution Time (usec.).....38.0
Object Code Bytes.....16

A DVI instruction divides the value in the accumulator by the immediate value of the DVI instruction, then writes the quotient back into the accumulator.

```
     A022     A035     14892                              A012
----(GET)---(MUL)---(DVI)-------------------------------(PUT)----
```

In this rung, the GET instruction writes the value of A022 into the accumulator. The MUL instruction multiplies the value of A035 by the value in the accumulator (A022) then writes the product back into the accumulator. The DVI instruction divides the value in the accumulator (A022 x A025) by 14892, then writes the quotient back into the accumulator. The PUT instruction writes the value in the accumulator ((A022 x A025)/14982) into variable A012. Each time this rung is executed, the value of A022 is multiplied by the value of A035, this product is divided by 14892, then the quotient is written into variable A012.

5.7 Logic Instructions

Logic instructions are always part of an input group in a rung. They perform their operations directly on the value in the accumulator.

Shift instructions move the bits in the accumulator left or right, 1 to 15 times.

The binary to BCD and BCD to binary instructions convert the value in the accumulator between 4 digit BCD values and 16 bit binary values.

The BCD logic instructions require a relatively large amount of memory and execution time. They should be used sparingly and only when needed.

Immediate Value
-(SLL)-

SLL (Shift Logically Left)

Keyboard Entry Code.....70
Possible Immediate Values..01-15
Execution Time (usec.).....16.0
Object Code Bytes.....6

The SLL instruction shifts the value in the accumulator logically left the number of times given by its immediate value.

The accumulator is a 16 bit register. In a shift left operation, all 16 bits are shifted left. The least significant bits fill with zeroes and the most significant bits are lost. The accumulator can be shifted 1 to 15 times. For example,

0001 0010 0000 1111

shifted 8 times left is

0000 1111 0000 0000

See the SLR instruction on the next page for an example of using the SLL instruction.

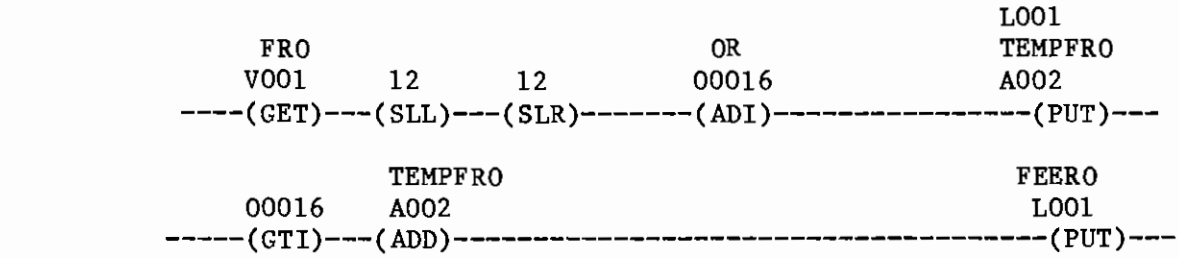
Immediate Value
-(SLR)-

SLR Shift Logically Right

Keyboard Entry Code.....71
Possible Immediate Values..01-15
Execution Time (usec.).....16.0
Object Code Bytes.....6

The SLR shifts the accumulator logically right the number of times given by its immediate value.

The accumulator is a 16 bit register. In a shift right operation, all 16 bits are shifted. The most significant bits fill with zeroes and the least significant bits are lost. The accumulator can be shifted 1 to 15 times.



In this example, the SLL and SLR instructions are used to properly adjust the value read from the feedrate override switch. The V001 variable is written into the accumulator by the GET instruction. The accumulator is shifted left 12 times, then right 12 times. This strips off the 12 most significant bits of V001, leaving the four lower bits. The result is put into A002 for temporary storage. For example,

V001 = 0010 0100 1100 1001

SLL 12 times = 1001 0000 0000 0000

SLR 12 times = 0000 0000 0000 1001 = A002

The last rung writes 16 into the accumulator with the GTI instruction. The ADD instruction adds A002 to the accumulator. The sum is put in L001, the value that the CNC reads for feedrate override.

16 + A002 = 0000 0000 0001 0000 + 0000 0000 0000 1001

L002 = 0000 0000 0001 1001 = 25

This procedure sends the feedrate override to the 8400 MP and Bandit III. If you are using the non-sequential switches, the value 16 is added to the four least significant bits of the value read from the switch. This tells the CNC to decode this value. 8400 controls that do not use non-sequential binary switches can load the four lower bits from the switch directly into L001.

-(BTD)-

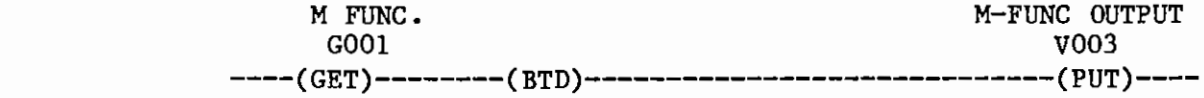
BTD
(Binary To BCD)

Keyboard Entry Code.....79
Possible Binary Values to Convert....0 - 9999
Execution Time (usec.).....132.0
Object Code Bytes.....66

The BTD instruction converts the value in the accumulator from a binary value to a binary coded decimal (BCD) value. After conversion, the result is written back into the accumulator.

The accumulator is a 16 bit register. Every group of four bits can assume a BCD value, 0 to 9. So, BCD values range from 0 to 9999.

	BCD		BCD
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001



In this rung, the GET instruction writes the value of G001 into the accumulator. This is the binary value of the current M-code. The BTD instruction converts this value from 16 bit binary to BCD. The PUT instruction writes the BCD value to the V003 output variable.

If an M23 is active, the accumulator contains this value after the GET instruction:

0000 0000 0001 0111

After the BTD instruction, the accumulator contains:

0000 0000 0010 0011

which is BCD 23.

PAL allows 5 digit entries for most immediate values. The largest number that could be represented in BCD format, however, is 9999. Remember, the accumulator is limited to 16 bits, or four groups of four bits.

-(DTB)-

DTB (BCD To Binary)

Keyboard Entry Code.....78
Possible BCD Values to Convert.....0 - 9999
Execution Time (usec.).....37.2
Object Code Bytes.....40

The DTB instruction converts the value in the accumulator from a BCD value to a binary value. After conversion, the result is written back into the accumulator.

The DTB instruction is useful when you have to convert BCD input values to binary before they are sent to the CNC.

```
                                                    CURRENT TOOL
     V002        8         8                           L008
----(GET)----(SLL)----(SLR)----(DTB)---------------(PUT)----
```

In this rung, the GET instruction writes the V002 variable into the accumulator. V002 contains the BCD code for the tool position that is currently seated. This value is in the lower 8 bits of V002. Assume a 16 position turret is being used. If the turret is at station 16, the accumulator contains:

0000 0000 0001 0110 (BCD)

The shift instructions strip off the unused bits, then the DTB instruction converts this value to a binary number. The accumulator now contains:

0000 0000 0001 0000 (binary)

This value is put into L008, the PAL to CNC variable for the tool that is currently seated.

PAL allows 5 digit entries for most immediate values. The largest number that could be represented in BCD format, however, is 9999. Remember, the accumulator is limited to 16 bits, or four groups of four bits.

5.8 Timer Instruction

Timer instructions are:

- always part of an output group in a rung
- edge triggered by the false to true change in preconditions of the rung

There are 16 timers that you can use in PAL. The timer instructions address these timers.

The time that a timer is on is given by an immediate value with the timer on, or retriggerable timer on, instructions. The immediate value for time can be in the range 0 to 65535, where each count stands for 10 ms. of time.

Important: PAL only executes once every 40 msecs. A timer set for less than 40 msecs. will time out before the next execution of PAL.

While a timer is on, its state corresponds to a logic 1. When the timer is off or timed out, its state corresponds to a logic 0. PAL can examine the state of timers with examine on and examine off instructions.

Timers are turned on or off according to the preconditions of the rung they are in. The preconditions must go from false to true for a timer instruction to have an effect. So, if you have a rung that controls a timer, you must control the preconditions of the rung to make the timer do what you want. A rung with a timer that goes from false to true and never goes false again is probably useless.

When a timer is retriggered, the time out of the timer begins again according to the amount of time given by the specified immediate value.

Timer
-(TON)-

TON (Timer On)

Keyboard Entry Code.....20
Possible Timers.....T001-T016
Possible Values For Time...00001-65535 (10 ms. per count)
Execution Time (usec.).....12.8
Object Code Bytes.....24

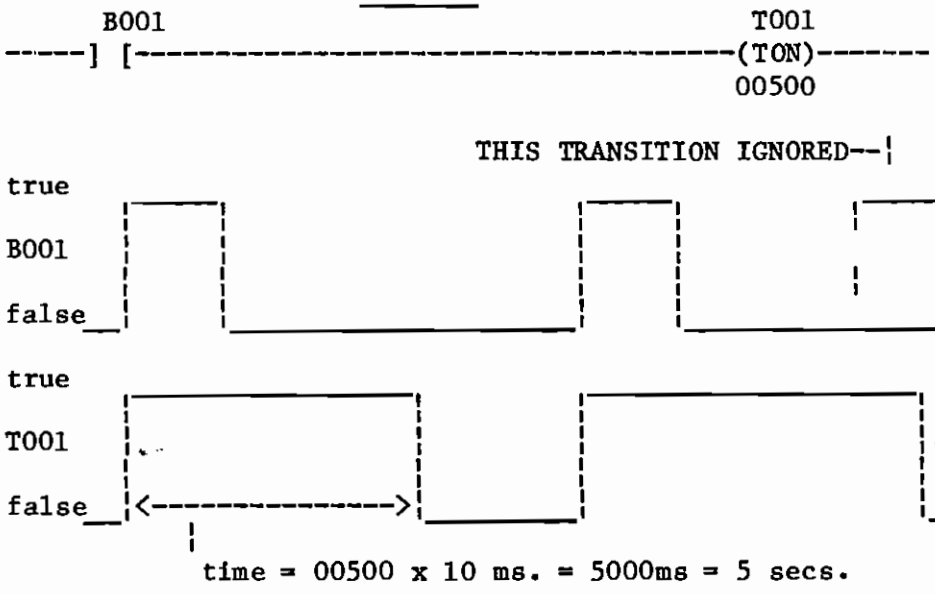
When rung preconditions go from false to true, the TON instruction turns its addressed timer on (true = 1) for the amount of time indicated by the immediate value for time. This means that timers are false to true edge triggered. Time is measured in units of 10 ms. An immediate value of 65535 (655.35 secs. the decimal point is assumed) turns on the timer for approximately 10 minutes and 55 seconds.

Initially, the timer is false (=0). This corresponds to a timed out condition. When rung preconditions go from false to true, the timer is set true (=1). The timer will remain true until "time" has elapsed or until turned off with a TOF instruction.

Important: The TON instruction is not "retriggerable." If rung preconditions go from false to true while the timer is on, the transition is ignored. If a timer is stranded in a GO-TO loop and not executed, it may not have seen a false to true transition. Without a false to true transition the counter will not reset and restart.

For all timers	T001 -[ ]-	Indicates Timer is active	T001 -[/]-	Indicates Timer is not active or has timed out

EXAMPLE



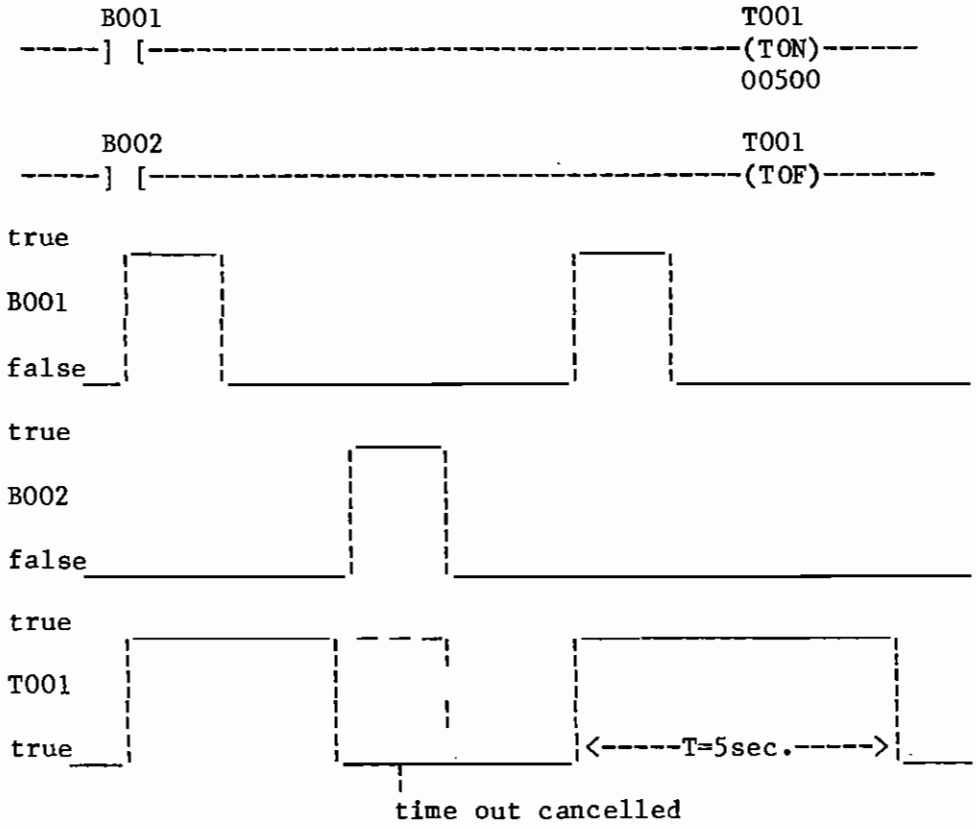
Timer
-(TOF)-

TOF (Timer Off)

Keyboard Entry Code.....21
Possible Timers.....T001-T016
Execution Time (usec.).....4.8
Object Code Bytes.....9

When rung preconditions go from false to true, the TOF instruction resets its addressed timer to off (false = 0). This corresponds to a timed out condition. As long as rung preconditions remain true, the addressed timer is held false (=0).

EXAMPLE



Timer
-(RTO)-

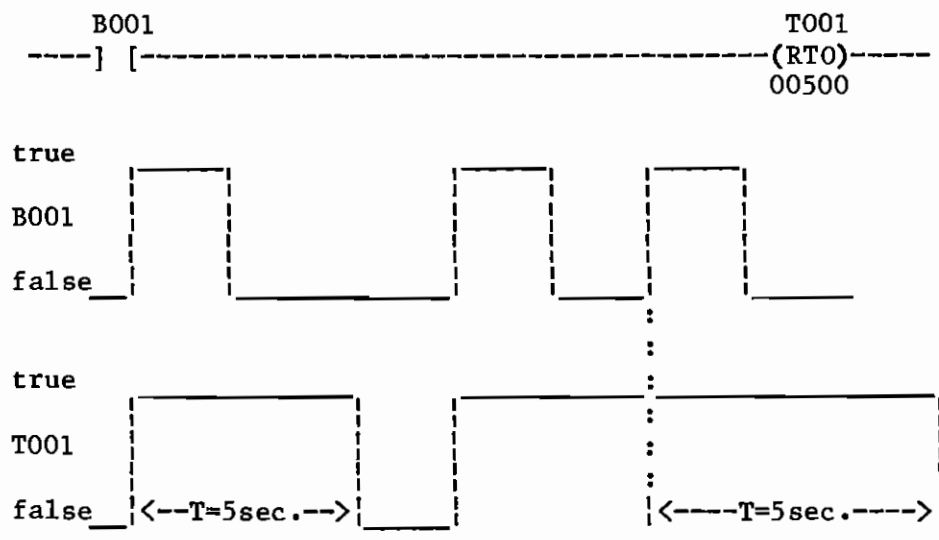
RTO (Retriggerable timer on)

Keyboard Entry Code.....22
Possible Timers.....T001-T016
Possible Values for Time...00001-65535 (10 ms per count)
Execution Time (usec.).....12.8
Object Code Bytes.....23

When rung preconditions go from false to true, the RTO instruction turns its addressed timer on for the amount of time indicated by the immediate value for time. This means that retriggerable timers are false to true edge triggered. Time is measured in units of 10 ms. A value for time of 65535 (655.35 secs.) corresponds to approximately 10 minutes and 55 seconds.

Initially, the timer is false (=0). This corresponds to a timed out condition. When rung preconditions go from false to true the the timer is set true (=1), and remains true until "time" has elapsed. If an rung preconditions changes from false to true again before "time" has elapsed, the timer will begin the "time" interval again until "time" elapses.

EXAMPLE



Timer
-(RTR)-

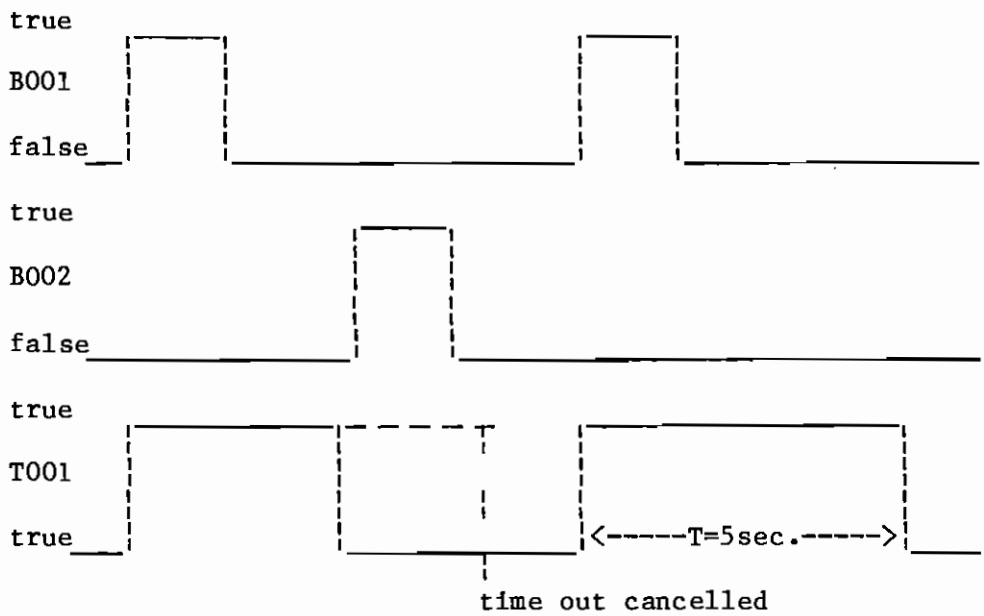
RTR
(Retriggerable
Timer Reset)

Keyboard Entry Code.....23
Possible Timers.....T001-T016
Execution Time (usec.).....4.8
Object Code Bytes.....9

When rung preconditions go from false to true, the RTR instruction resets its addressed timer to off (false = 0). The RTR instruction resets a timer that has been turned on with a RTO instruction. As long as rung preconditions remain true, the addressed timer is held false (=0).

EXAMPLE

```
     B001                                        T001
-----] [----------------------------------------(RTO)------
                                                 00500

     B002                                        T001
-----] [----------------------------------------(TOF)------
```



SPACE for NOTES

5.9 Counter Instructions

Counter instructions are:

- always part of an output group in a rung
- edge triggered by the false to true change in preconditions of the rung

There are 16 counters that you can use in PAL. The counter instructions address these counters.

Counters have a range from 0 to 65535 and "rollover" at 0 and 65535. In other words, the next count up from 65535 is 0, and the next count down from 0 is 65535.

Counters are initially 0 unless you load them with a value by using a PUT instruction. The value of a counter can be loaded into the accumulator using a GET instruction that addresses the counter.

Counters count up or down according to the instruction you use and according to the preconditions of the rung they are in. The preconditions must go from false to true for a counter instruction to have an effect. So, if you have a rung that controls a counter, you must control the preconditions of the rung to make the counter do what you want. A rung with a counter that goes from false to true and never goes false again is probably useless.

```
    Counter
    -(CTU)-
```

CTU (Count Up)

Keyboard Entry Code.....30
Possible Counters.....C001-C016
Execution Time (usec.).....11.6
Object Code Bytes.....17

When rung preconditions change from false to true, the CTU instruction adds one count to its addressed counter. The counter does not increment when rung preconditions change from true to false.

The CTU instruction normally counts up from "0", but the counter can be initialized with any value by using a PUT instruction that addresses the counter.

The maximum value for a counter is 65535. The next count up from this value writes zero into the counter, and more counting starts from zero.

```
    END                                          COMP.CYCLE
    N036                                           C001
----]/[-----------------------------------------(CTU)----
```

In this rung, each time N036 (the PAL to CNC flag for program end) goes from a true (=1) to false (=0) transition, the examine off instruction makes rung preconditions go from false to true. The CTU instruction increments C001 by 1 count for each transition false to true.

You could use the value of C001 as part of a message.

```
    COMP.CYCLES
       C001                                        L037
------(GET)-------------------------------------(PUT)----
```

L037 is the PAL to CNC variable that is shown with the M037 message.

```
                                                CYCLE COMP.
                                                   M037
-------------------------------------------------(OTE)----
```

This last rung constantly displays the message CYCLE COMP. along with the value of L037 (C001). This indicates how many cycles are complete.

Counter
-(CTD)-

CTD (Count Down)

Keyboard Entry Code.....31
Possible Counters.....C001-C016
Execution Time (usec.).....11.6
Object Code Bytes.....17

When rung preconditions go from false to true, the CTD instruction subtracts one count from the current value in the counter. The counter does not decrement when rung preconditions change from true to false.

The minimum count is zero. Another count down from this value causes the counter to return to 65535, and count down from there.

```
     00050                                                C003
---(GTI)-------------------------------------------------(PUT)---

     B025                                                 C003
----] [-------------------------------------------------(CTD)---
```

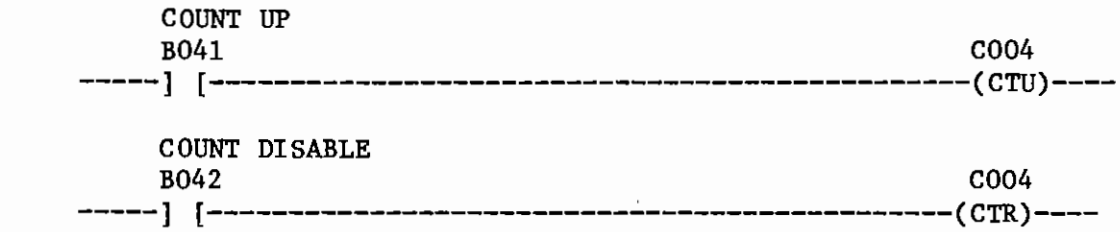
The first rung writes the value 00050 into the accumulator with the GTI instruction. The PUT instruction initializes the counter C003 with this value. In the second rung, a false (=0) to true (=1) transition of B025 makes rung preconditions go from false to true. Counter C003 decrements by 1 count for each B025 false to true transition.

Counter
-(CTR)-

CTR
(Counter Reset)

Keyboard Entry Code.....32
Possible Counters.....001-C016
Execution Time (usec.).....4.8
Object Code Bytes.....10

When rung preconditions go from false to true, the CTR instruction resets the value of its addressed counter to zero. As long as rung preconditions remain true, the counter value remains zero.



In the first rung, each time B041 goes from false (=0) to true (=1), C004 increments by 1 count. In the second rung, when B042 goes from false (=0) to true (=1) C004 is reset to zero. C004 is held at zero for as long as B042 is true (=1), regardless of the state of B041. If B042 goes false (=0), C004 can begin counting up again, once for each false to true transition of B041.

A.0 What is PAL?

PAL (Programmable Application Logic) is a software interface between your machine tool and the 8400 control for non-axis motion functions.

Control software reads servo inputs (direction and speed) and controls outputs to servo loops. PAL reads other inputs (limit switches and relays) and controls other outputs (solenoids).

PAL differs from part programming. Part programs are generally done by the end user of the control after it has been integrated with the machine. Part programming controls the machine functions according to the software parameters set in AMP and the hardware functions determined by the PAL.

PAL differs from AMP (Adjustable Machine Parameters). PAL defines logical relations among inputs and outputs. AMP specifies values for various machine parameters without defining logical relations. AMP configures control software machine parameters (maximum speeds and soft overtravels). PAL is used to define machine/control logic.

PAL programming is generally done by the machine tool builder. PAL is necessary before part programs can be successfully executed. PAL typically handles S, T, and M word decoding, turret rotation, and spindle rotation.

A.1 Ladder Diagrams

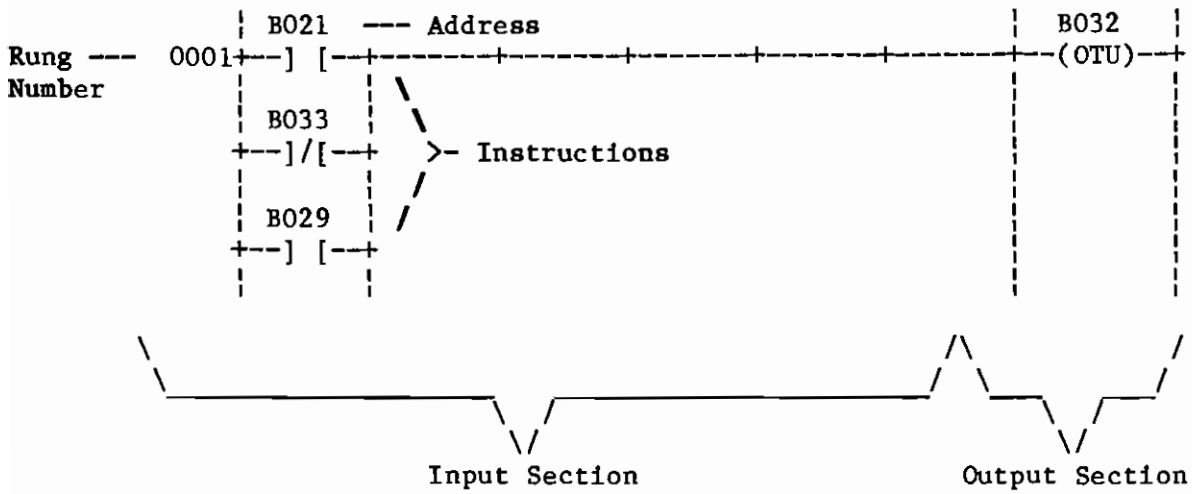
Instead of wiring relays to perform machine logic, you can program a ladder diagram that defines the logic, then store this program in your control. The control executes the PAL program, which defines relations between:

- inputs to PAL from the machine tool, operator panel, part program, or peripheral devices
- outputs from PAL to the machine tool, operator panel, part program, or peripheral devices

PAL's ladder diagramming scheme is similar to conventional relay logic diagrams, except that PAL instructions replace the traditional electronic symbols.

A ladder diagram consists of one or more separate and complete rungs which can logically use the input data it receives to control an output or a set a condition.

Input and output elements of a rung are made up of their own specific instructions. Each PAL instruction has its own unique symbol and performs a specific function. Refer to chapter 3.)



You can combine input elements to form both AND and OR conditions. An AND operation is one in which all input elements in series must be true in order for the output element to perform its function. In an OR operation only one input element, out of a group of parallel inputs, need be true for the output element to respond. You can program AND and OR conditions in many different combinations in a single rung.

PAL executes a ladders rungs sequentially. One program scan iteration is one complete execution cycle of all the rungs in the program except those skipped by GO-TO instructions.

The PAL program reads through the ladder diagram and sets the condition of the rungs to a true state when it gets to each rung. As each input instruction is executed, it can, depending on its' status, make the rung false.

Once a rung becomes false it remains false until the next program scan. The last instruction in each rung is an output instruction which controls its' bit or word address according to the rungs' condition - either true or false.

During operation, the control follows a definite sequence of operation with respect to PAL:

1. Read inputs PAL examines the input signals from the machine tool, paramacros, and the part program. It relays the input information to the control's processor.

2. Execute PAL program

   The processor monitors the data and executes the pre-programmed logic operations based on the inputs' status.

3. Set outputs accordingly

   The processor generates the appropriate responses and transmits the signals to the machine tool or a peripheral device.

By continually repeating this sequence, the PAL continually updates outputs according to the latest input data.

B.0 Using the Development Program with the Kaypro

This section gives you a series of brief steps for using the development program, PAL.COM and XREFLAD.COM, to generate PAL programs.

The detailed explanation of how to use PAL.COM is given in chapter 5. This outline of the steps is a preview of that chapter.

It is good practice, if this is your first time through 8400 PAL, to use the example program in chapter 4 for your first attempt. It will give you experience with the Kaypro, and help you learn the structure and function of PAL programs.

Developoment Steps

1. Power up the Kaypro and insert the Main Disk in drive A and the Data Disk in drive B (see chapter 4, Organize Your Development Disks). Type in "[PAL] and press [Return]. Press the CAPS LOCK key. PAL.COM only recognizes capital letters.

2. Type "I" to select I/O Assignments. You will use parallel, non-inverted I/O and use 7408 buffers for outputs. Set reader or punch enable, if desired. Define each port as input, output, or unused. Define port pairs for V variables, if desired. Type Return to get back to the Main Menu.

3. Type "E" to enter the edit mode. Write in the PAL program from chapter 4. Make a list of all messages used by M flags. Reserve status message M028 for the PAL program part number.

4. Once you've input the ladder, assign messages with the message display.

5. Save the PAL source code (PAL source code can be saved at any time and should be save frequently in case of power loss).

6. Compile the PAL source code and correct any compilation errors. Save the source code again if changes are made. You will have to recompile after saving the changed source code.

7. Download the PAL object code to the 8400 for testing. We strongly recommend that initial testing be done on a test fixture with switches and lights only. Extreme caution is advised when testing on a "live" machine.

8. When the PAL program works properly, save the corrected PAL source code again, and then recompile.

9. Enter "Hex I/O" mode. Set the baud rate at 1200. Enter terminal mode and initialize the prom programmer. Exit terminal mode and select object code output.

10. Burn object code EPROM's for permanent PAL program installation in the 8400.

11. Make a print out of the PAL program using "PRINT", and make a cross reference of the program using XREFLAD.COM.

C.0 Equipment Requirements

Before you can implement a PAL program with your 8400 Series control you need the following equipment:

- a Kaypro 2 personal computer - to develop, edit, compile and download the PAL ladder diagram
- the PAL software on a 5 1/4 disk from Allen-Bradley containing:
  - the PAL Development Software (part of the Executive file invoked by typing "PAL [RETURN]")
  - the XREFLAD file (also part of the Executive file) which is invoked by typing " XREFLAD [RETURN]"
- a PROM burner - to put the compiled source program onto the proms that will go into your 8400 Series control
  - the requirements for the PROM burner to use are on page 2-3 and 2-4
- the EPROMS to hold the PAL
  - the 8400 GP, MP, and Robot controls use 2716 EPROM
  - and the 8400 GLC control uses 2764 EPROMs
- a printer to give you the hard copy you will need for quick reference and for troubleshooting PAL.
  - The suggested printer is an Epson Model Number FX-80 Dot Matrix Printer, or equivalent.
- Series 8400 control to put the EPROMS into. This will allow you to perform a carefully prepared and closely monitored test of the PAL before it is used in production.
- Cables to interconnect these components.
- I/O Hardware and Cable.
  - I/O Board
  - I/O Modules

C.1 Connector Cables

You will need two cables to connect the Kaypro to the PROM burner and 8400 CPU board.

- One cable from the Kaypro 2 (connector J4) to the PROM burner.
- A separate cable from the Kaypro 2 (connector J4) to the revision A CPU module of the 8400 control. See the table below:

Table C-1

8400 Hardware Revision	CPU Connector	Kaypro 2 Connector
A	CN-14	J4
B	P3	J4

Important: The Kaypro 2 can be connected to the revision B CPU Module connector P3 on the 8400 control by making a cable for the following connections:

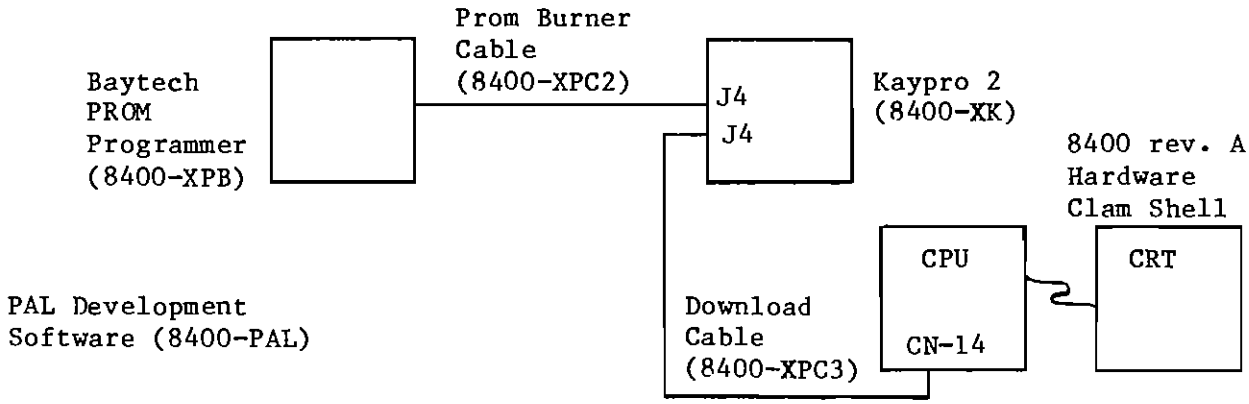
Table C-2

ConnectorP3 P3 on CPU	Connect To	D Shell Connector on Kaypro
Pin 13	connect to	7
Pin 1	connect to	1
Pin 3	connect to	2
Pin 5	connect to	3
		Connect pins 6, 8 and 20 together on this end.

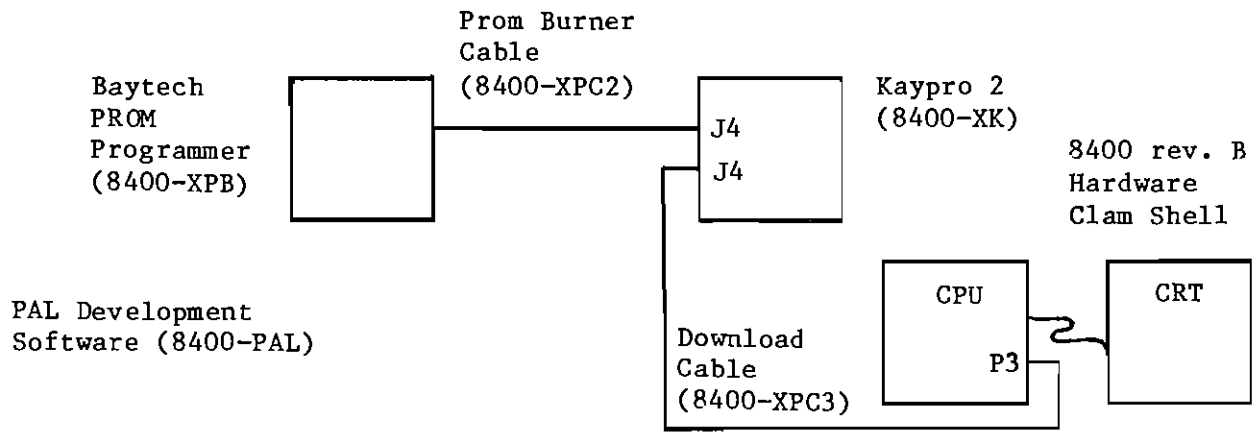
C.3 Connecting PAL Development Components

This diagram shows the connections between components for PAL program development for revision A and revision B hardware on 8400 Series controls.

Revision A Hardware Cabling Diagram



Revision B Hardware Cabling Diagram



C.4 Prom Programmer Requirements

The following is a list of the requirements for any prom programmer to be used with the PAL program:

- An RS232 port must be available with one of the following baud rates: 150, 300, 600, 1200, 2400, 4800, or 9600.
- Transmit and receive (via the RS232 port) Intel Hex Formatted data.
- Burn 2716 EPROMS (for use with the 8400 GP, MP or R software).
- Burn 2764 EPROMS (for use with the 8400 GLC).

We recommend using the Bay Tech Associates 953B Prom Burner, (A-B Cat. No. 8400-XPB) or equivalent to meet your 8400 needs.

Table 1

Compiler and Editor Error Codes

Error	Corresponding Message	Stop Compiling	Cause of Error
01	No "BEG" statement	Y	First character in ladder source is not BEG.
02	No ladder file exists	Y	"END" statement immediately follows "BEG" statement in ladder source file.
03	Incomplete input group	Y	I) "END" statement found while decoding an input group.
		N	II) "SEQ" statement found while decoding an input group.
04	No output found on this sequence	Y	I) "END" statement found immediately after an input.
		N	II) "SEQ" statement found immediately after an input.
05	Unrecognized sequence	Y	I) Character immediately following "BEG" statement is neither "SEQ" or "END".
		Y	II) Character immediately following an output is neither another output nor "SEQ" or "END".
06	Arithmetic accumulator not yet initialized	N	I) Arithmetic function requested but arithmetic accumulator has not been loaded.
		N	II) Arithmetic output requested but arithmetic accumulator has not been loaded.
07	Arithmetic found within input group	N	Arithmetic load or arithmetic function requested within a group.
08	Input element found immediately following an output element	N	Input element found after output element in sequence.

Table 1 (Cont.)

Compiler and Editor Error Codes

Error	Corresponding Message	Stop Compiling	Cause of Error
09	Output element found within input group	N	Output requested while decoding an input group.
10	Undefined subrung	N	A new rung of an input group is started before the last rung has been completed. i.e. BST is not preceeded by BND for all but the first BST of an input group.
11	Syntax error in branching control statements	N	BND element found outside of any input branch groups.
12	Undefined element	N	I.) Invalid mnemonic II.) Mnemonics in invalid order III.) "END" statement missing
13	Multiple definition label	N	Same value declared as a label more than once.
14	Undefined label	N	Value referred to by a GTO element but never declared as a label.
15	I/O element not declared in I/O descriptor file	N	Input or Output instruction requires use of port which has not been declared for use.
16	Variable address out of bounds	N	Variable associated with element is out of bounds of variable type.
17	Invalid operand type	N	Operand type associated with element is not accessible using this element.
18	Attempt to write to input port	N	Output element tries to write to an input port.
19	Attempt to use dedicated port	N	Output element tries to write to a dedicated port.

Compiler and Editor Error Codes

Error Editor Only	Corresponding Message	Stop Compiling	Cause of Error
20	Sequence picture limits exceeded	N	I) more than 5 outputs attempted II) more than 4 BST's in an input group III) attempt to enter input elements into output area.
21	Group width limits exceeded.	N	Attempt to enter too many input elements in a rung of a previously defined group.
22	Construction error	N	Attempt to follow BST immediately by BND
23	I/O port pair not defined in I/O descriptor file	N	Input or Output instruction requires use of port pair which has not been defined.

Table 2

Intel Hex Format

RECORD MARK
This is always a colon ":" indicating the start of new record

RECORD LENGTH
Two ASCII characters representing a hex number indicating how much DATA will be in this record. The number 10 indicates there will be 16 DATA bytes in this record. The number 00 indicates an END RECORD.

START ADDRESS
Four ASCII characters representing a hex number indicating where the first byte of DATA is located in memory.

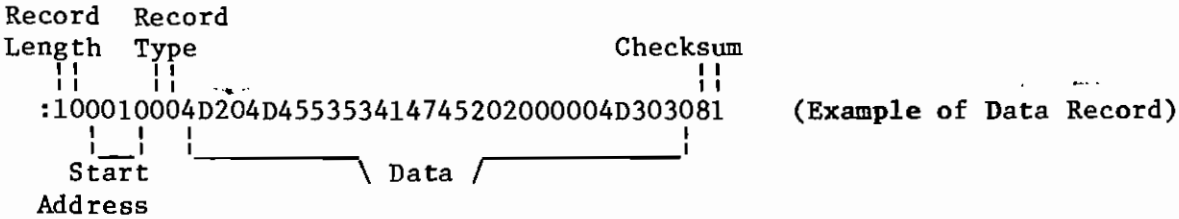
RECORD TYPE
Two ASCII characters indicating the type of record. There are just two types of records. These are:

00 = Data Record
01 = End Record

DATA
Two ASCII characters represent one byte of DATA in hexadecimal format.

CHECKSUM
Two ASCII characters represent one CHECKSUM byte in hexadecimal. This number may be calculated by adding the hexadecimal numbers represented by the two ASCII characters from the RECORD LENGTH to the last DATA byte. Then calculate the two's complement of this number. Using the least significant 8 bits of this is CHECKSUM.

Example of Data record CHECKSUM:
10 + 00 + 10 + 00 + 4D + 20 + 4D + 45 + 53 + 53 + 41
+ 47 + 45 + 20 + 20 + 00 + 00 + 4D + 30 + 30 = 37F
The Two's Complement of 37F = C81
The least significant 8 bits results in 81.



:00000001FF (Example of End Record)

Table 3

Parallel Interface with Normal I/O

Outputs:

Ladder Diagram Source Code Elements using R001 to R120 Variables	8400 Logic Level Output Device = 8400-Z0 Input Device = 8400-Z1	Solid State Relay
+-(OTE)-+	LOGIC 0 LOW	OUTPUT "ON"
+-(OTD)-+	LOGIC 1 HIGH	OUTPUT "OFF"
+-(OTL)-+	LOGIC 0 LOW	OUTPUT "ON"
+-(OTU)-+	LOGIC 1 HIGH	OUTPUT "OFF"

Inputs:

Solid State Relay	8400 Logic Level Output Device = 8400-Z0 Input Device = 8400-Z1	Ladder Diagram Source Code Elements using R001 to R120 Variables
INPUT "OFF"	LOGIC 1 HIGH	+--[ ]--+ Element will be "FALSE"
INPUT "OFF"	LOGIC 1 HIGH	+--[/]--+ Element will be "TRUE"
INPUT "ON"	LOGIC 0 LOW	+--[ ]--+ Element will be "TRUE"
INPUT "ON"	LOGIC 0 LOW	+--[/]--+ Element will be "FALSE"

A "1" in a "V" Variable will be a LOGIC 0 LOW Level on the corresponding 8400 Input or Output. If it is a Solid State Device output, it will be "ON". If it is a Solid State Device input, the input will be "ON".

A "0" in a "V" Variable will be LOGIC 1 HIGH Level on the corrsponding 8400 Input or Output. If it is a Solid State Device output, it will be "OFF". If it is a Solid State Device input, the input will be "OFF".

Modifying PAL REV. K4 Y22

Newer Kaypro computers, Those that use graphics characters show the following problems when when using the PAL.COM program (copyright Oct. 18, 1983, rev.Y22):

1. Graphic characters are turned on during editing and compiling.

2. Download to the 8400 does not work. This is also a problem with the old Kaypro's when downloading to the 8400GP.

3. Transfer of Intel Hex data to Bay Technical Associates PROM programmer works only at 300 baud.

What the Modified Program Will Do

The modified program will fix all three problems listed above. The modified program will also have a new name; PALK4.COM. You'll see the following differences in the operation of PALK4.COM:

- The revision for the program, shown on the Main Menu page, will be changed from Rev.Y22 to PK4.Y22.

- After you press [D] to start a download to the 8400, the system will immediately begin the download. It will not request that you press [S] to begin the download.

- After the download is completed the system will automatically begin the execution of the PAL program in 8400 RAM. It will not request that you press [S] to begin execution of RAM PAL.

The Proceedure

Use the following proceedure to modify PAL.COM (copyright Oct. 18, 1983, rev.Y22). After modifying PAL.COM, it will work with any Kaypro as of 5/14/84.

1. After powering up the Kaypro, put CP/M with DDT into disk drive A. You'll see the CP/M sign-on message on the CRT.

2. Put PAL.COM, copyright Oct. 18, 1983, rev. Y22 into disk drive B.

3. Type [DDT B:PAL.COM] [RETURN]
   DDT should sign on with:

```
DDT VERS 2.2
NEXT PC
9000 0100
```

4. Make the necessary changes to the program by typing the the the information in quotes (" "):

```
"S4D9D[RETURN]"
4D9D 52 "20[RETURN]"
4D9E 20 "20[RETURN]"
4D9F 3D "20[RETURN]"
4DA0 20 "20[RETURN]"
4DA1 52 "20[RETURN]"
4DA2 45 "20[RETURN]"
4DA3 54 "20[RETURN]"
4DA4 52 "20[RETURN]"
4DA5 59 "20[RETURN]"
4DA6 20 "20[RETURN]"
4DA7 4C "20[RETURN]"
4DA8 49 "20[RETURN]"
4DA9 4E "20[RETURN]"
4DAA 4B "20[RETURN]"
4DAB 1B ".[RETURN]"
```

5. Type what is in quotes.

```
"S4DAF[RETURN]"
4DAF 4F "20[RETURN]"
4DB0 52 "20[RETURN]"
4DB1 20 "20[RETURN]"
```

6. Type what is in quotes.

```
"S158[RETURN]"
158 C2 "C3[RETURN]"
159 B9 ".[RETURN]"
```

7.Type what is in quotes.

```
"S50B4[RETURN]"
50B4 11 "0[RETURN]"
50B5 06 "0[RETURN]"
50B6 4F "0[RETURN]"
50B7 CD "0[RETURN]"
50B8 9B "0[RETURN]"
50B9 08 "0[RETURN]"
50BA 3E ".[RETURN]"
-
```

8. Type what is in quotes.

```
"S514C[RETURN]"
514C 11 "0[RETURN]"
514D 21 "0[RETURN]"
514E 4F "0[RETURN]"
514F CD "0[RETURN]"
5150 9B "0[RETURN]"
5151 08 "0[RETURN]"
5152 CD "0[RETURN]"
5153 BB "0[RETURN]"
5154 51 "0[RETURN]"
5155 B7 "0[RETURN]"
5156 C2 "0[RETURN]"
5157 A0 "0[RETURN]"
5158 51 "0[RETURN]"
5159 3E ".[RETURN]"
```

9. Type what is in quotes.

```
"S57A9[RETURN]"
57A9 80 "0[RETURN]"
57AA B0 ".[RETURN]"
```

10. Type what is in quotes.

```
"S57AE[RETURN]"
57AE 40 "0[RETURN]"
57AF 07 ".[RETURN]"
```

11. Type what is in quotes.

```
"S727[RETURN]"
727 52 "50[RETURN]"
728 45 "4B[RETURN]"
729 56 "34[RETURN]"
72A 2E ".[RETURN]"
```

12. Type what is in quotes.

```
"S508B[RETURN]"
508B 3E "11[RETURN]"
508C 01 "06[RETURN]"
508D 0E "4F[RETURN]"
508E FF "CD[RETURN]"
508F CD "9B[RETURN]"
5090 50 "08[RETURN]"
5091 52 "3E[RETURN]"
5092 B7 "01[RETURN]"
5093 11 "0E[RETURN]"
5094 5D "FF[RETURN]"
5095 4D "CD[RETURN]"
5096 C2 "50[RETURN]"
5097 7B "52[RETURN]"
5098 51 "B7[RETURN]"
5099 11 "11[RETURN]"
509A A3 "5D[RETURN]"
509B 4E "4D[RETURN]"
509C CD "C2[RETURN]"
509D 98 "90[RETURN]"
509E 08 "51[RETURN]"
509F CD "0[RETURN]"
50A0 BB "0[RETURN]"
50A1 51 "0[RETURN]"
50A2 B7 "0[RETURN]"
50A3 C2 "0[RETURN]"
50A4 A0 "0[RETURN]"
50A5 51 "0[RETURN]"
50A6 3E ".[RETURN]"
```

13. Hold [CTRL] and press [C]. The Kaypro returns to the CP/M operating system.

14. Type "Save 143 B:PALK4.COM[RETURN]" Your edited patches will be saved to PALK4.COM in disk drive B.

To Run the New Program

The patched program is now ready to run. Make this program part of your Main disk as described earlier in section 4.

To access the new program for PAL program development type:

PALK4 [return]

Note that the name of the program has changed from PAL.COM to PALK4.COM.

Advanced PAL Example

Once you have planned and allocated the inputs and outputs for PAL, you can proceed with planning your PAL program. This is what we call "pre-programming." There are two aspects to pre-programming: flowcharting and sketching the ladder.

The first thing you should do in pre-programming is to make a flowchart of your PAL program. It gives you a base for a systematic approach to PAL programming, it gives structure to your PAL programs and allows you to add steps easily when they are required.

The example flowchart, given below, is for a full function ladder: the analog spindle control with Quickdraw Automatic Toolchanger ladder, P/N 251-100 76, which is the one of the standard ladders available for the 8400 MP/Bandit III CNC systems. Study this flowchart and try to get an idea of how 8400 PAL programs are structured.

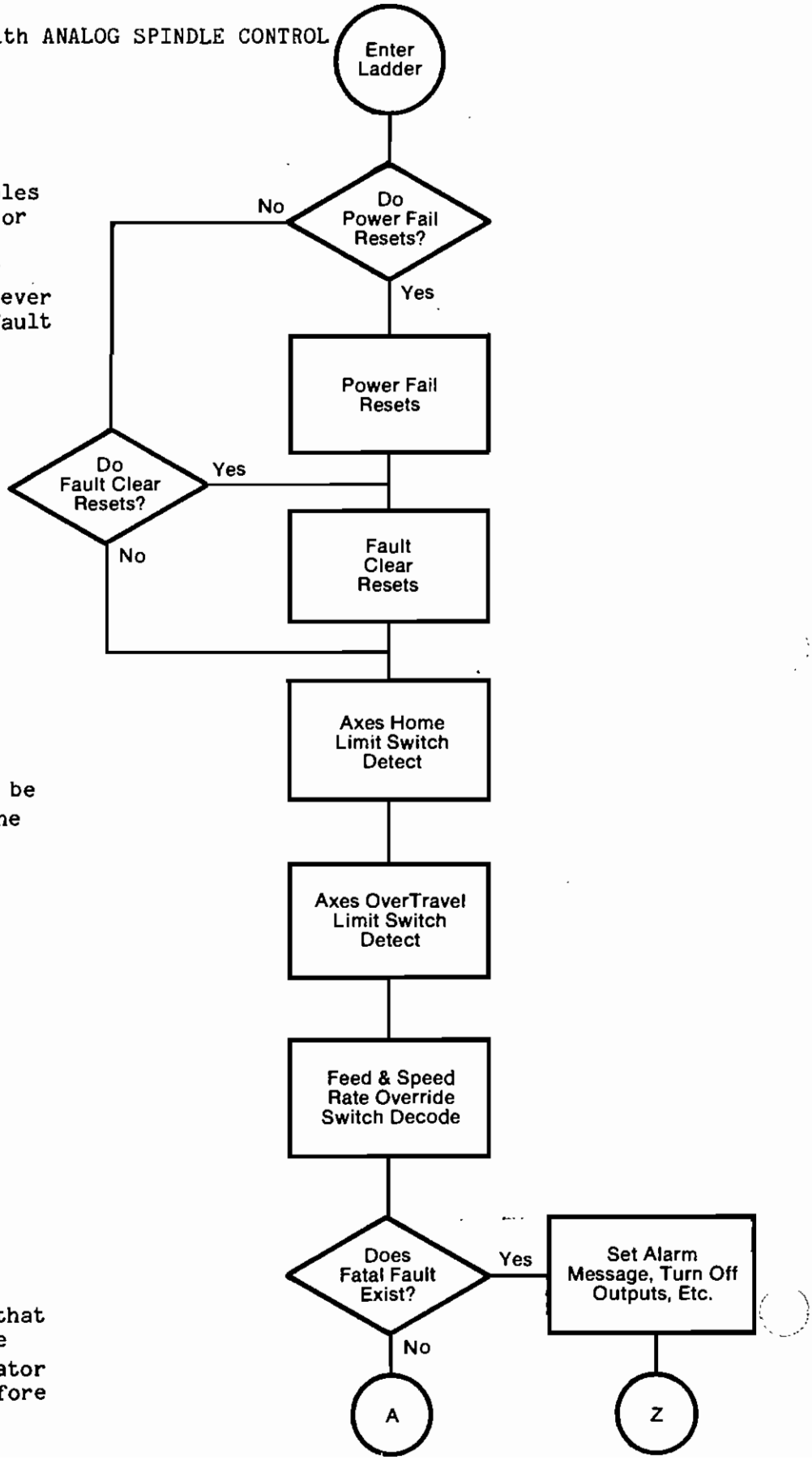
NOTE: One thing that affects 8400 PAL structure is the time is spent executing PAL and the flow of execution. Remember that 8400 PAL takes 2 to 6 ms. to execute, and the ladder is executed every 40 ms. You can't have a program that takes a lot of time. You cannot jump "backward" in PAL, and each execution of the ladder must flow through from beginning to end. Your program must copy the state of inputs, make required tests, do required actions, and be ready for the next pass. If an action is not complete on one pass through PAL (PAL may be waiting for a switch closure, a certain count, or a timed out timer), you must set flags to tell PAL where to begin again on the next pass.

Flowchart for 8400 PAL Program with ANALOG SPINDLE CONTROL

Some flags and variables may need to be reset or initialized only on power-up. Others may need to be reset whenever the CNC generates a Fault Clear signal.

The Home, overtravel, feedrate and spindle speed switches should be examined every time the ladder is executed.

A Fatal Fault is one that shuts down the machine and requires the operator to clear the fault before continuing.

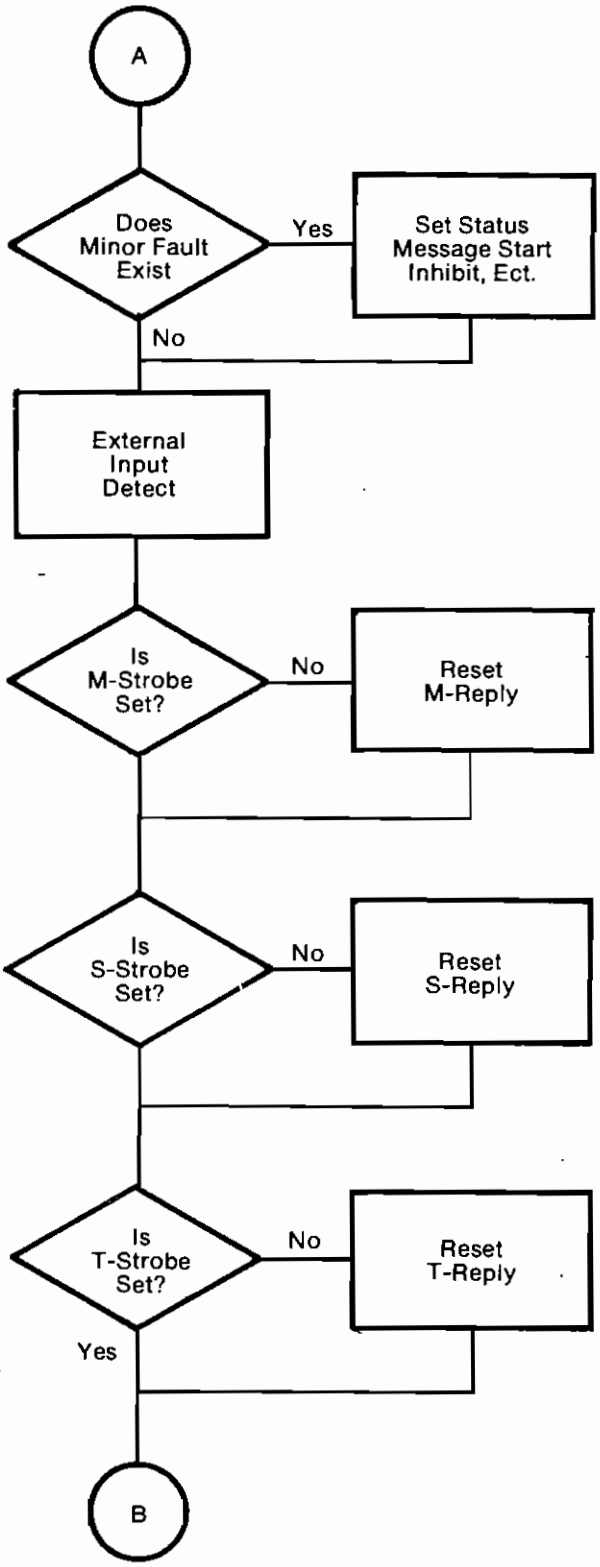


MISCELLANEOUS INPUT DETECT (A)

A Minor Fault does not stop the machine immediately. A warning message is displayed and the machine may be put in the Start Inhibit state.

External inputs include Remote Start/Stop, Feedhold, Start Inhibit, Keyboard Lockout, etc.

If the function strobe has gone away, then the function reply flag is reset.



ACTION DECODE ROUTINE (B)

If the Delay flag is set, then the rest of the PAL program is skipped.

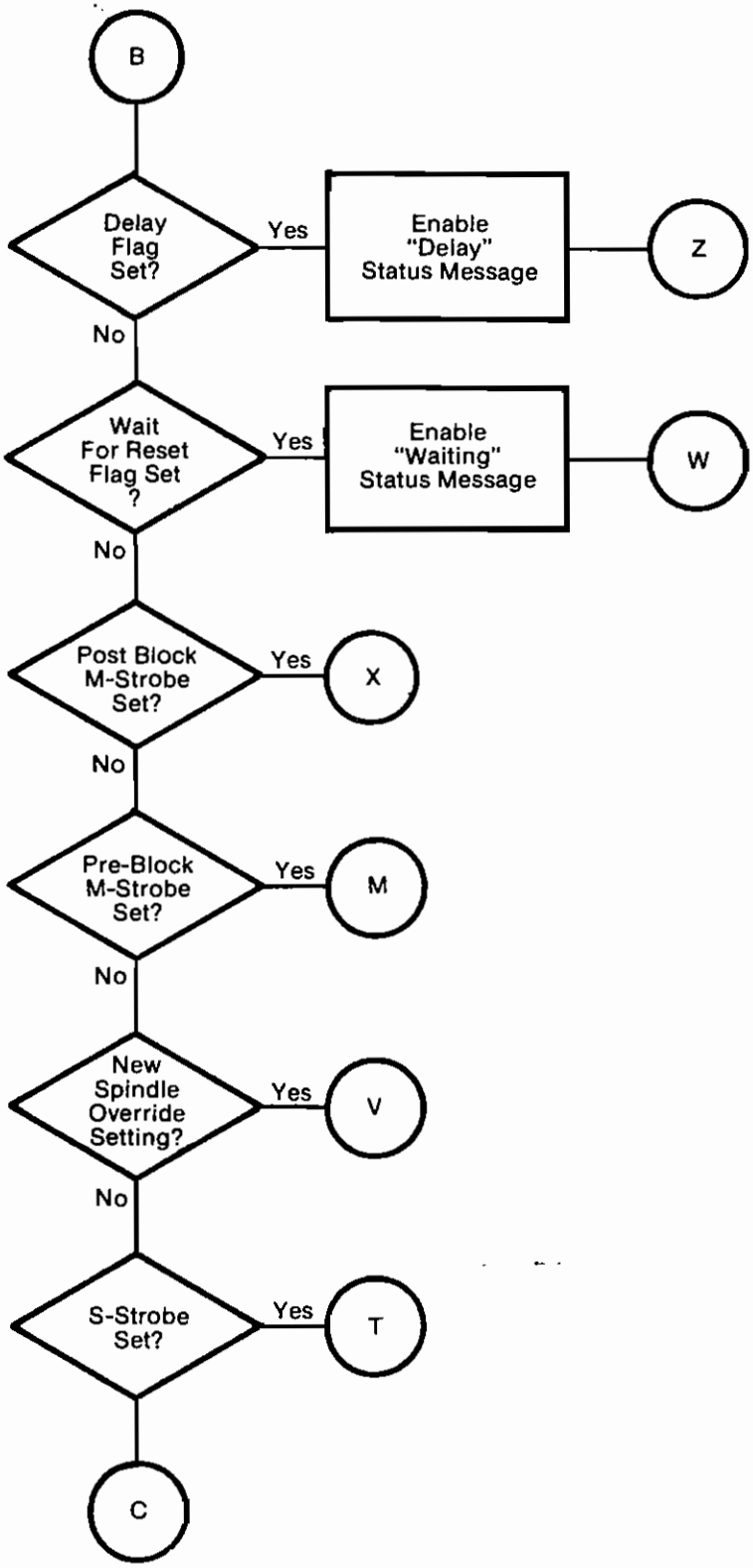
If the Wait for Reset flag is set, then jump to the Reset Detect routine.

If a post-move M-strobe is detected, then jump to the Post-Move routine.

If a pre-move M-strobe is detected, then jump to the Pre-Move routine.

If the spindle override switch has changed, then jump to the Spindle Override portion of the Spindle Speed routine.

If an S-strobe is detected then jump to the Spindle Speed routine.

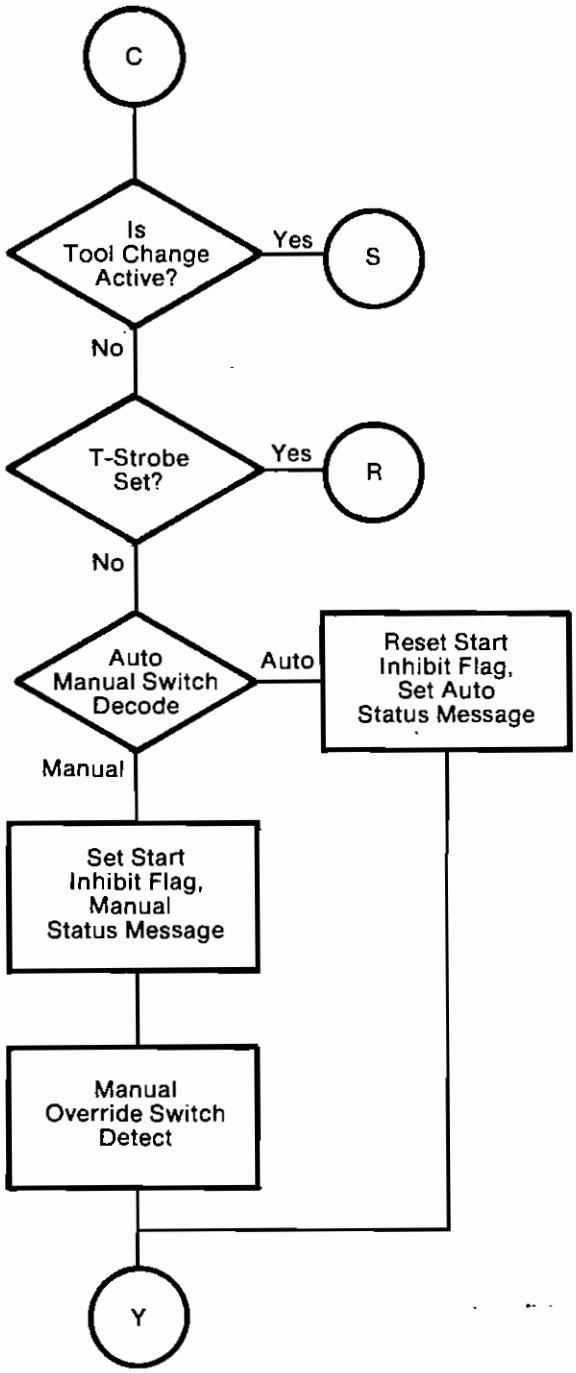


ACTION DECODE ROUTINE (continued) C

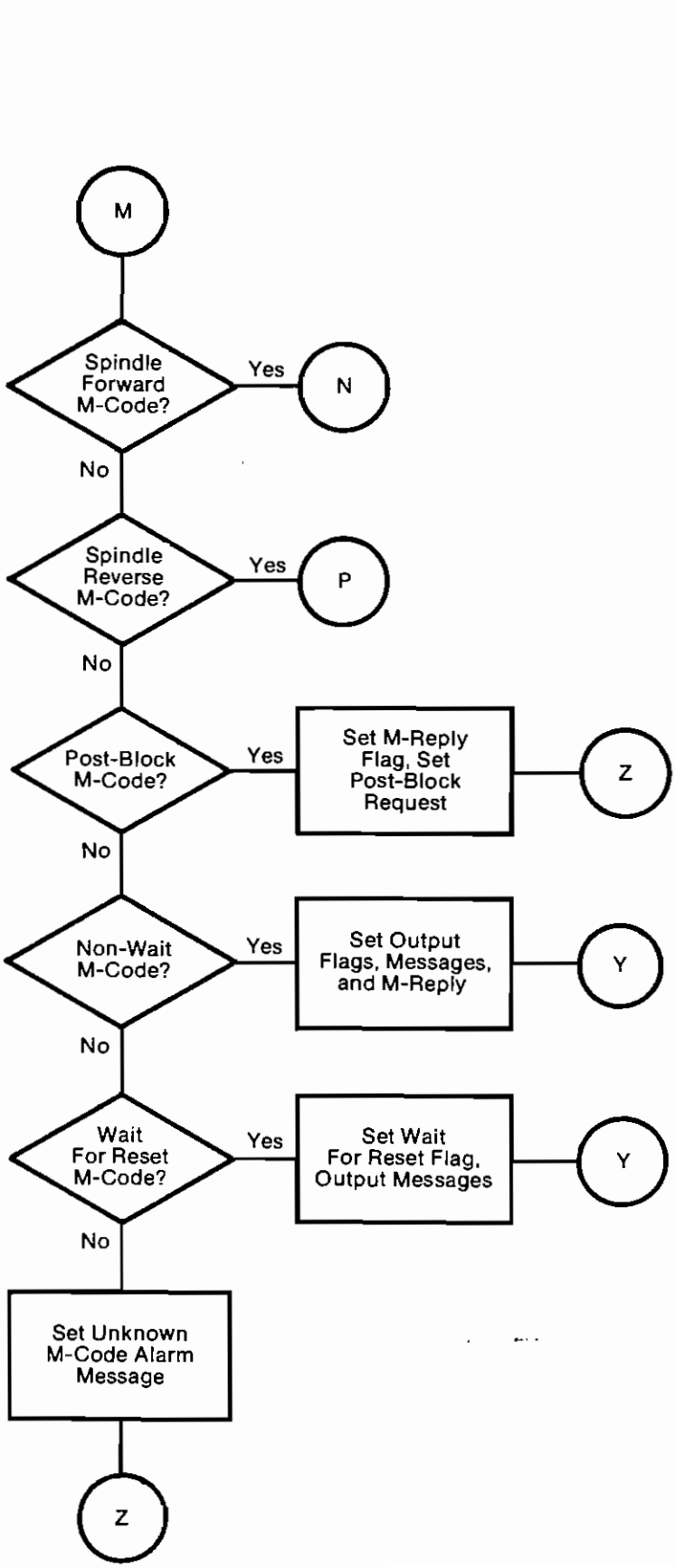
If a tool change is already in progress, then jump to Tool Position Check portion of the Tool Change routine.

If a T-strobe is detected, then jump to the Tool Change routine.

The setting of the AUTO/MANUAL switch will determine whether the operator is allowed to override the spindle and coolant. In AUTO, programs will run and the overrides are ignored. In MANUAL, programs cannot be started and the override switches are active.



PRE-MOVE M-FUNCTION DECODE



On M3, jump to the Spindle Forward routine.

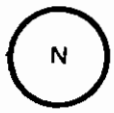
On M4, jump to the Spindle Reverse routine.

If the M-code is post-move, simply set the M-reply flag and exit the ladder (the 8400 MP and Bandit III do not require setting the post-move request flag).

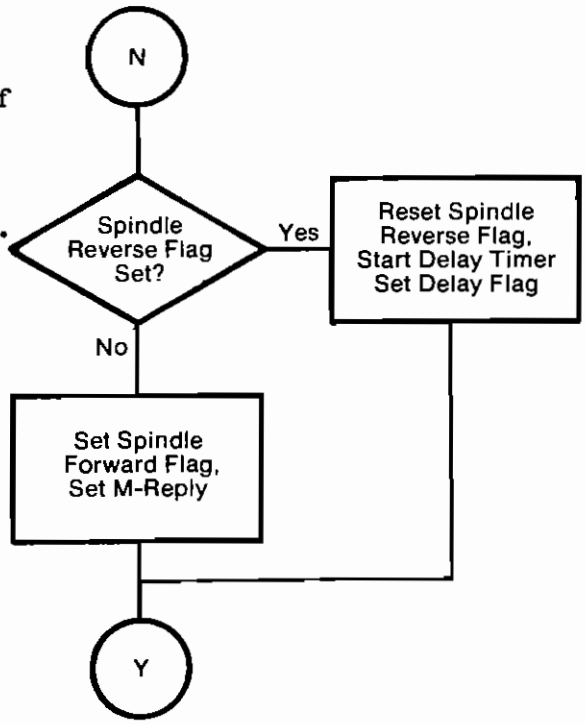
If the M-code does not require response, set the M-reply flag & the appropriate out flags, and jump to the Output Handling routine. If the M-code does require response, do not set the M-reply flag. Do set the Wait for Reset flag.

If the M-code is undefined, then an Alarm message should be used.

SPINDLE FORWARD ROUTINE

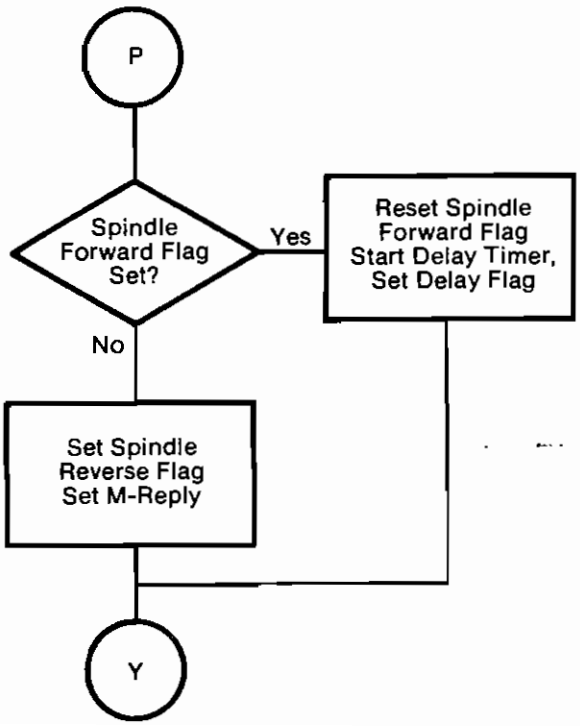


If the spindle is on reverse, then turn it off and wait before turning it on forward. The M-reply is not set until the function is complete.

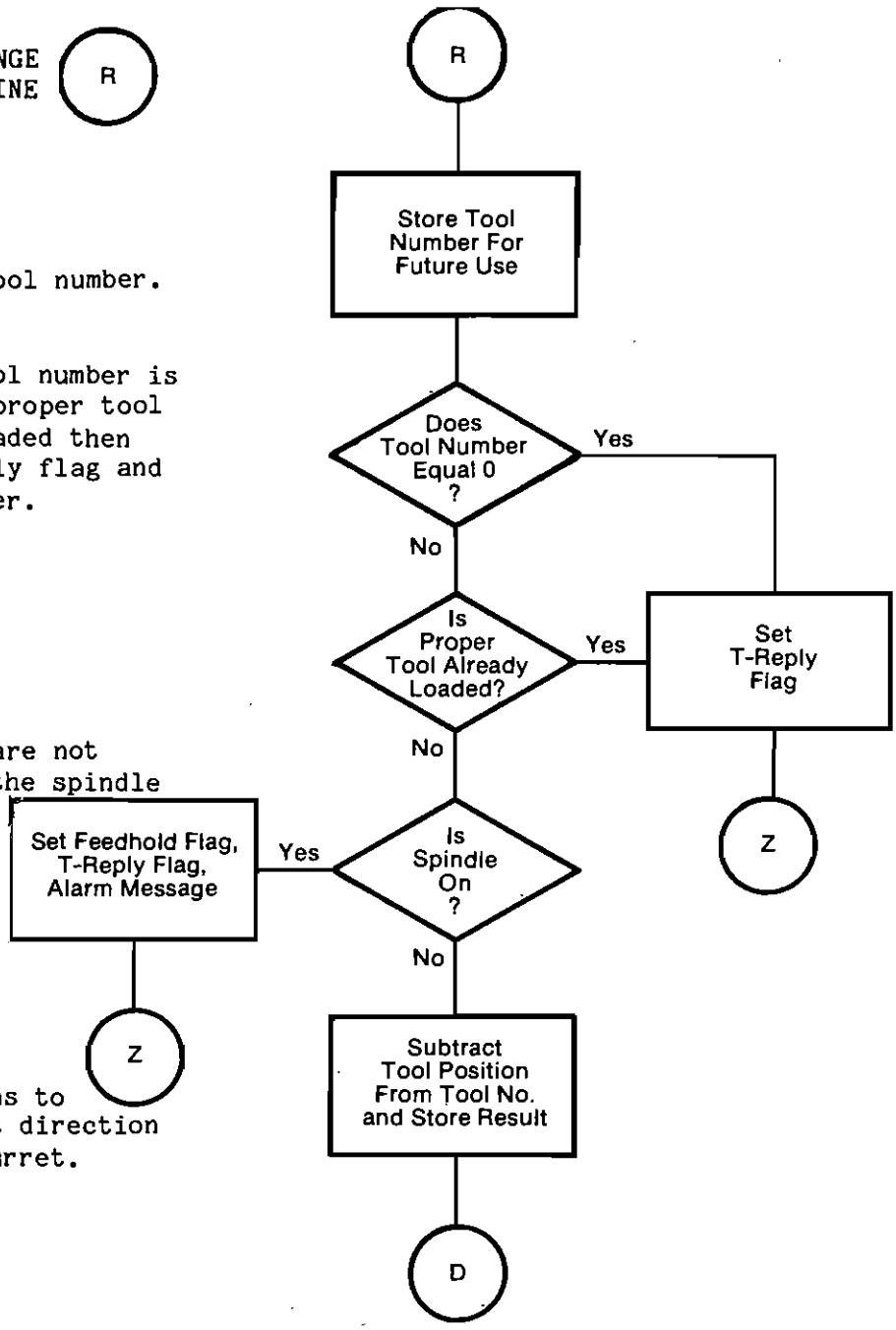


SPINDLE REVERSE ROUTINE

If the spindle is on forward, then turn it off and wait before turning it on reverse.



TOOL CHANGE ROUTINE



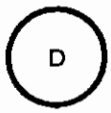
GET the new tool number.

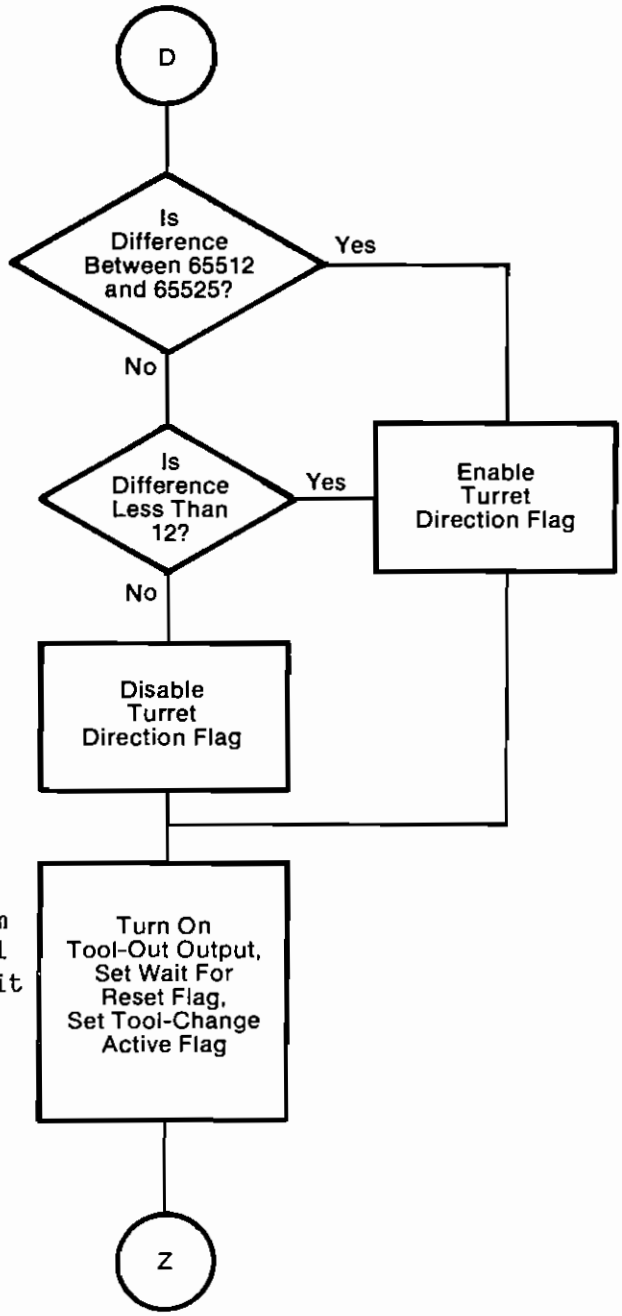
If the new tool number is zero, or the proper tool is already loaded then set the T-reply flag and exit the ladder.

Tool changes are not allowed with the spindle on.

Do calculations to determine what direction to turn the turret.

TOOL CHANGE ROUTINE (continued)

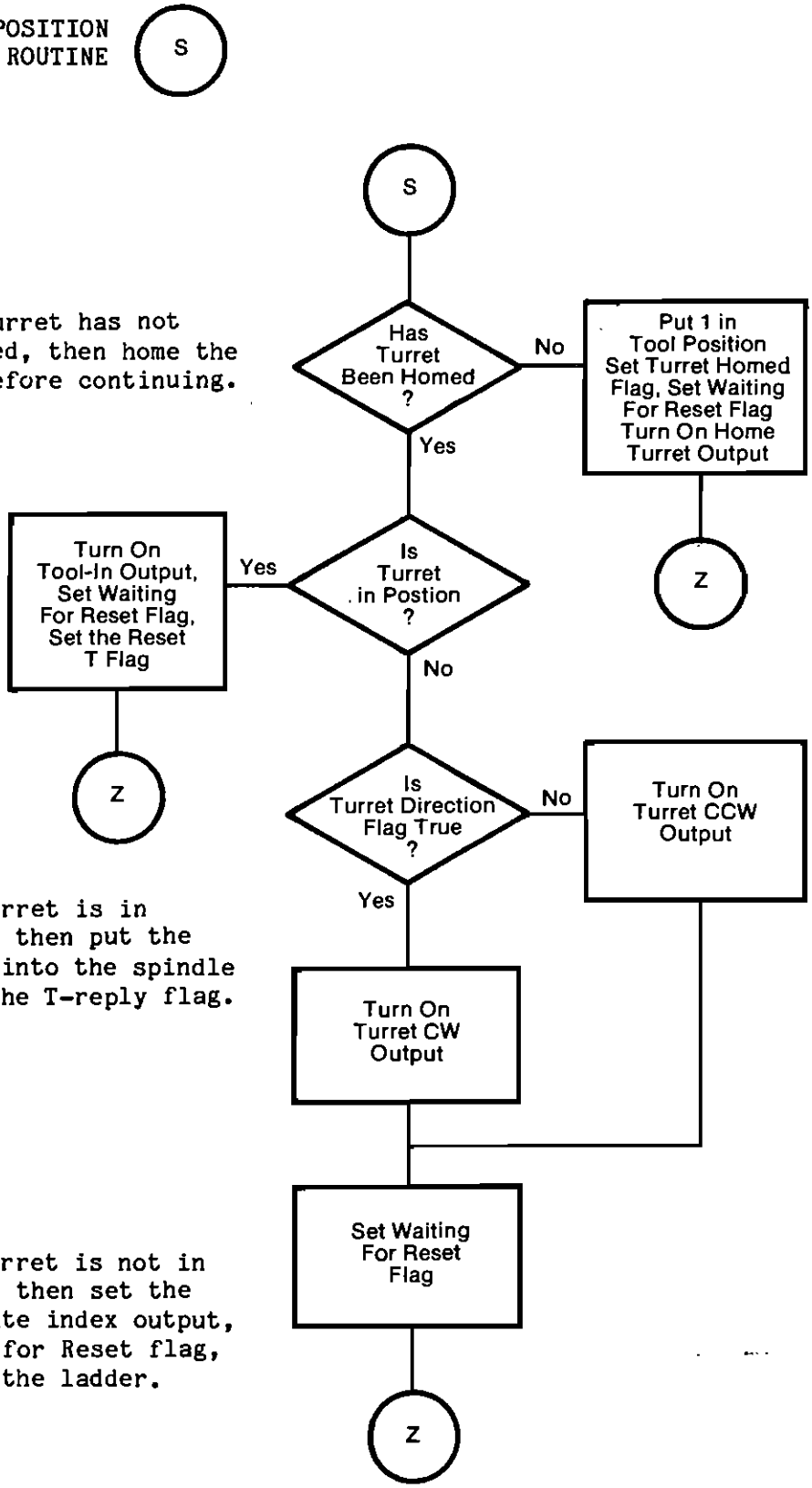




Turret direction calculation

Remove current tool from spindle and set the Tool Change Active flag. Exit the ladder. The next execution will jump to the Tool Position Check routine for the Action Decode routine.

TOOL POSITION CHECK ROUTINE

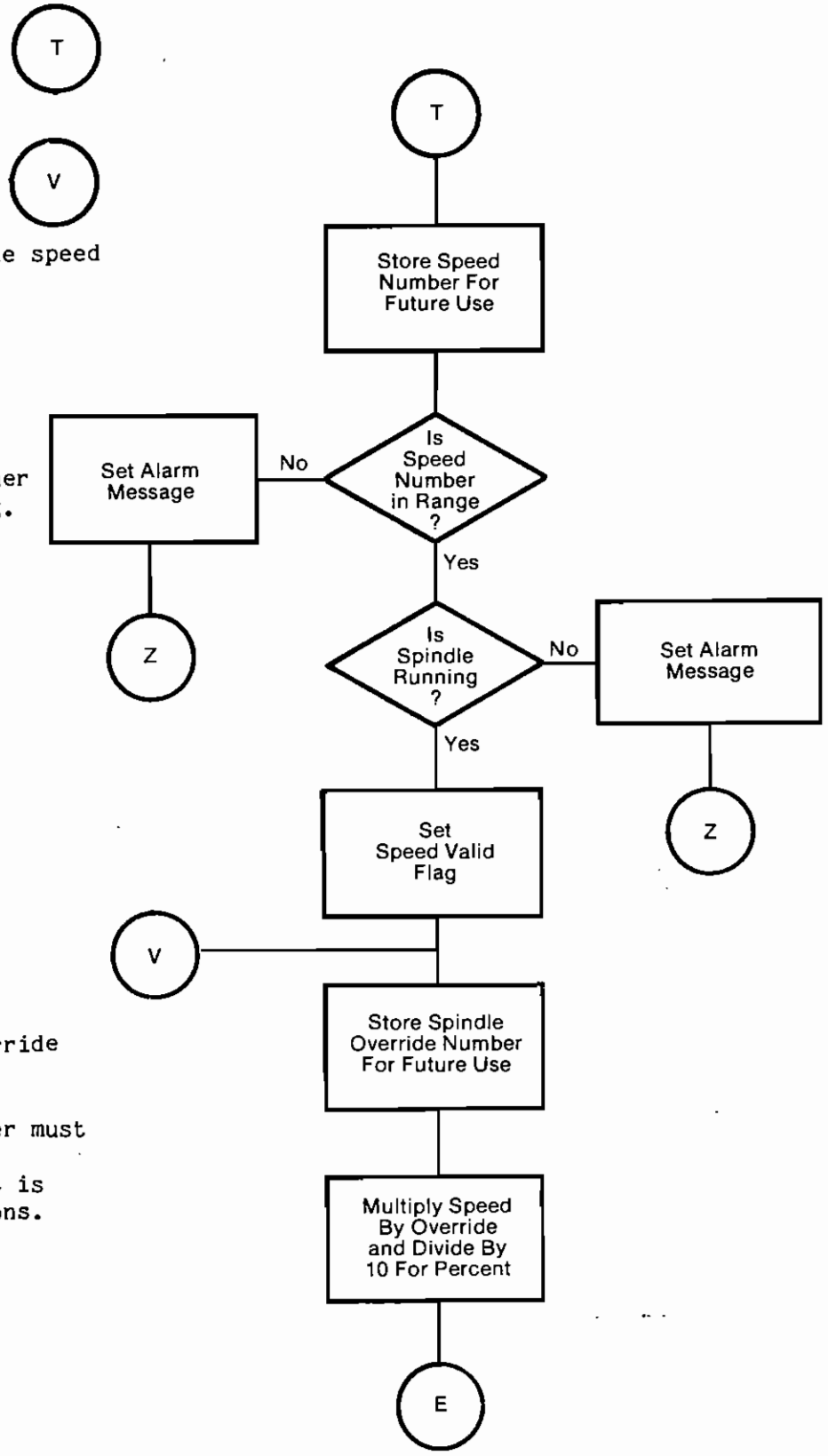


If the turret has not been homed, then home the turret before continuing.

If the turret is in position, then put the new tool into the spindle and set the T-reply flag.

If the turret is not in position, then set the appropriate index output, the Wait for Reset flag, and exit the ladder.

SPINDLE SPEED ROUTINE



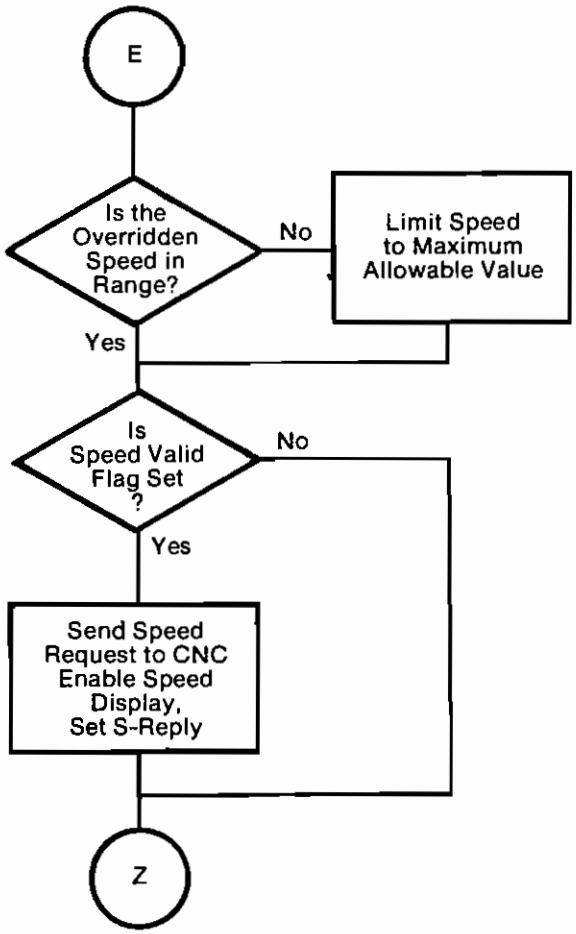
Spindle Speed Override Routine

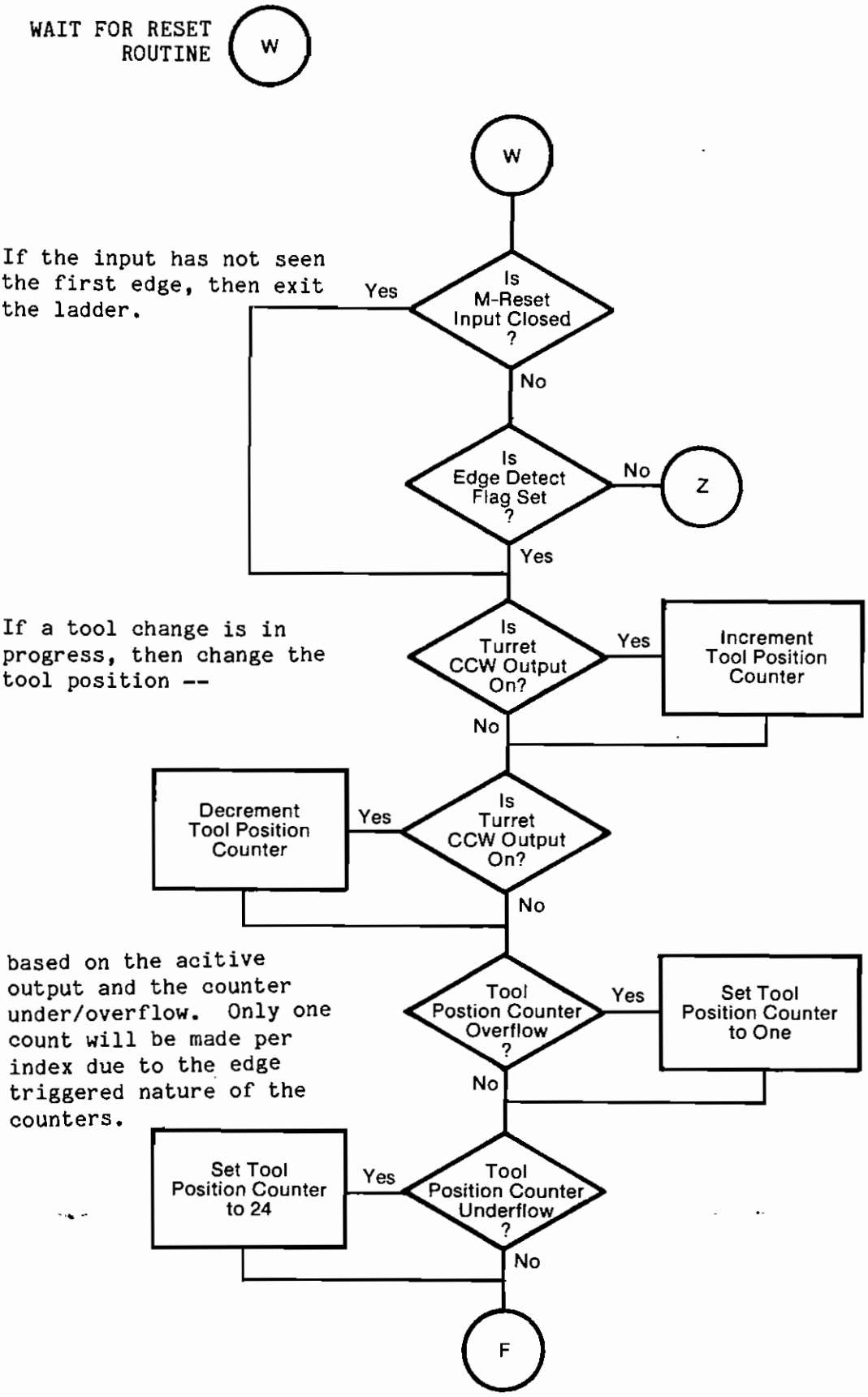
The override number must be stored in an A variable before it is used in calculations.

SPINDLE SPEED AND OVERRIDE ROUTINE (continued)

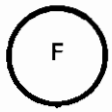
If use of the override switch violates the max or min RPM limits, then replace speed number with max or min allowable.

If the speed request is valid, then change the analog output & spindle speed display, and set the S-reply. Unless a speed request (S-code) has been executed, no change in analog output will occur due to the override switch changing.



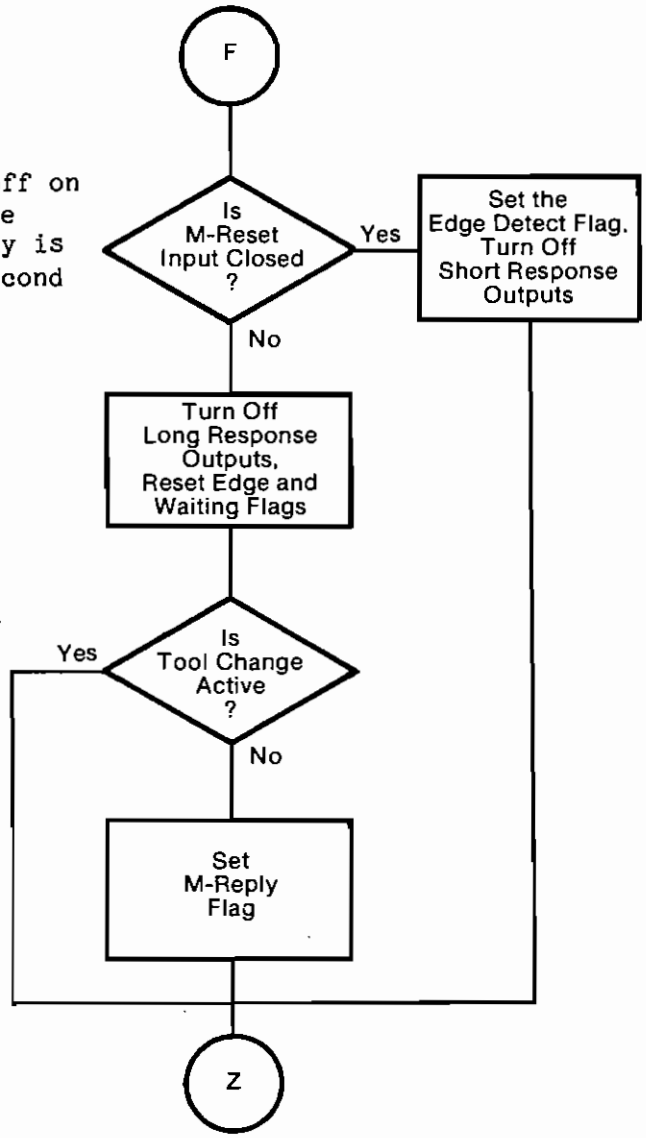

WAIT FOR RESET ROUTINE
W
W
If the input has not seen the first edge, then exit the ladder.
Is M-Reset Input Closed ?
Yes
No
Is Edge Detect Flag Set ?
No
Z
Yes
If a tool change is in progress, then change the tool position --
Is Turret CCW Output On?
Yes
Increment Tool Position Counter
No
Decrement Tool Position Counter
Yes
Is Turret CCW Output On?
No
based on the acitive output and the counter under/overflow. Only one count will be made per index due to the edge triggered nature of the counters.
Tool Postion Counter Overflow ?
Yes
Set Tool Position Counter to One
No
Set Tool Position Counter to 24
Yes
Tool Position Counter Underflow ?
No
F

WAIT FOR RESET ROUTINE (continued)

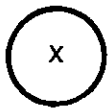


Short response type outputs are turned off on the first edge of the reset pulse. M-reply is not set until the second edge is detected.

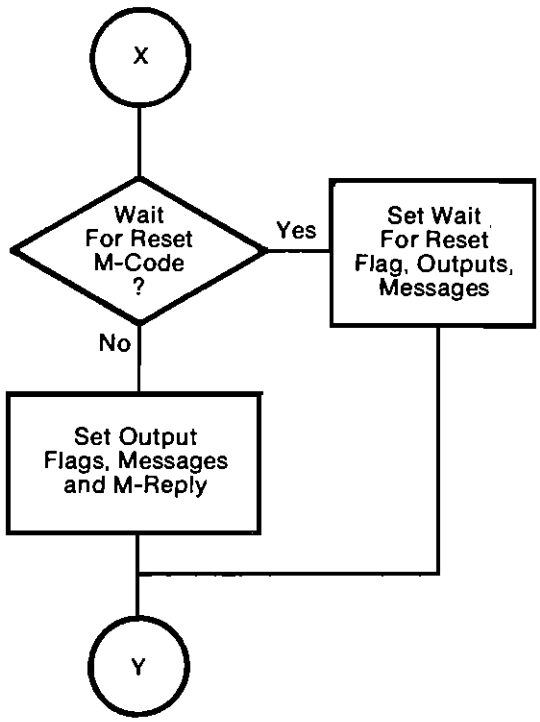
If a tool change is active, then M-reply should not be set.



POST-MOVE
M-FUNCTION
DECODE

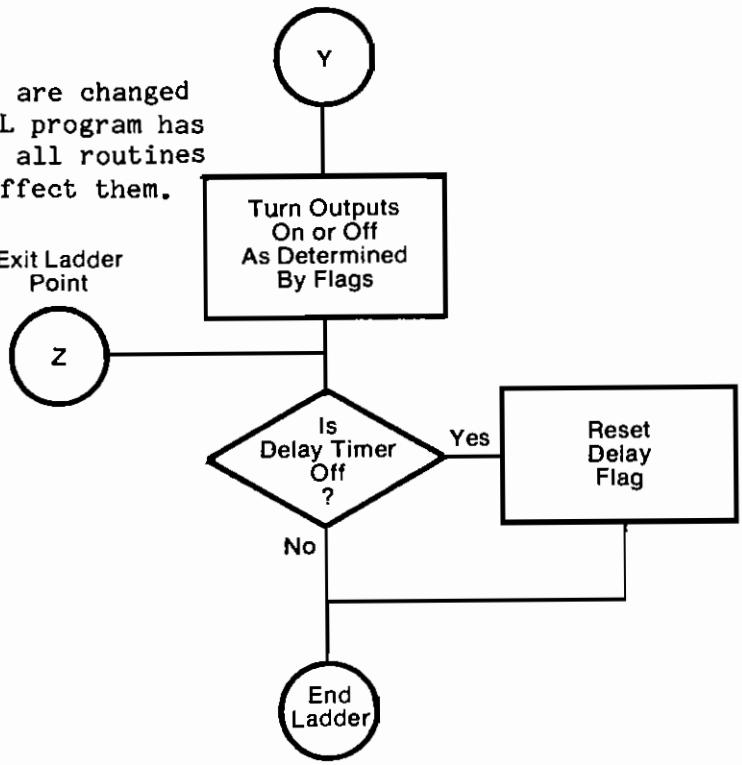


M-reply flag must be set for all M-functions with 8400 MP and Bandit III.



OUTPUT HANDLING ROUTINE

Real outputs are changed after the PAL program has gone through all routines that might affect them.



Sketching the Ladder

After you have a good flowchart for your ladder, you should sketch the ladder program.

Sketch the ladder the way you will enter it using the Kaypro.

Outline your use of R, B, N, and H flags, and assign mnemonics to each one. Also specify your use of V, L, and G variables and assign mnemonics to these as well. If you use counters or timers, assign mnemonics to these. Use the mnemonics in the sketch.

Write out your assignment of messages, M flags, their text and associated L variables (used with variable messages).

The example program that follows is a short ladder that implements several standard functions you will probably use in any 8400 PAL program:

- Fault Clear to PAL from the CNC
- Emergency Stop switch monitoring
- Feedrate Override switch monitoring
- Drives On
- Drives OK
- Axis move enable (clamps off)
- Home switch monitoring
- Overtravel switch monitoring
- M-functions (pre- and post-move)

This page tells you the I/O allocations that have been made. The system uses non-inverted parallel I/O. PAL ports 1 through 3 and 7 through 8 are inputs. Ports 10 through 12 and 14 & 15 are outputs. V variable V001 is made from ports 7 and 8. No parallel devices are used.

12-30-83 LADDER LIST: 8400 EXAMPLE PAL PROGRAM PAGE 1

SYSTEM USES NON-INVERTED I/O

SYSTEM USES PARALLEL I/O

I/O GROUP	DEFINED AS	PORT IS ON	CN #	R RANGE		I/O PORT#	PORT IS
1	IN	CPU	CN15	R001-R008	IC154	30H	A
2	IN	CPU	CN15	R009-R016	IC154	32H	B
3	IN	CPU	CN15	R017-R024	IC154	34H	C
4	UDFN	CPU	CN16	R025-R032	IC163	38H	A
5	UDFN	CPU	CN16	R033-R040	IC163	3AH	B
6	UDFN	CPU	CN16	R041-R048	IC163	3CH	C
7	IN	I/O	CN8	R049-R056	IC119	80H	A
8	IN	I/O	CN8	R057-R064	IC119	82H	B
9	IN	I/O	CN8	R065-R072	IC119	84H	C
10	OUT	I/O	CN10	R073-R080	IC158	90H	A
11	OUT	I/O	CN10	R081-R088	IC158	92H	B
12	OUT	I/O	CN10	R089-R096	IC158	94H	C
13	UDFN	I/O	CN5	R097-R104	IC84	B0H	A
14	OUT	I/O	CN5	R105-R112	IC84	B2H	B
15	OUT	I/O	CN5	R113-R120	IC84	B4H	C

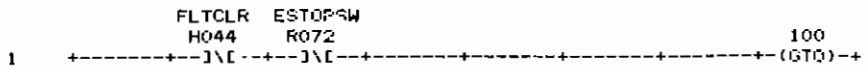
V NUMBER	HI	LO
V001	7	8
V002	0	0
V003	0	0
V004	0	0
V005	0	0
V006	0	0
V007	0	0
V008	0	0

PAPER TAPE READER IS NOT INSTALLED
PAPER TAPE PUNCH IS NOT INSTALLED
CASSETTE IS NOT INSTALLED

This page gives you the message print out. Note which messages are assigned and the text for each message.

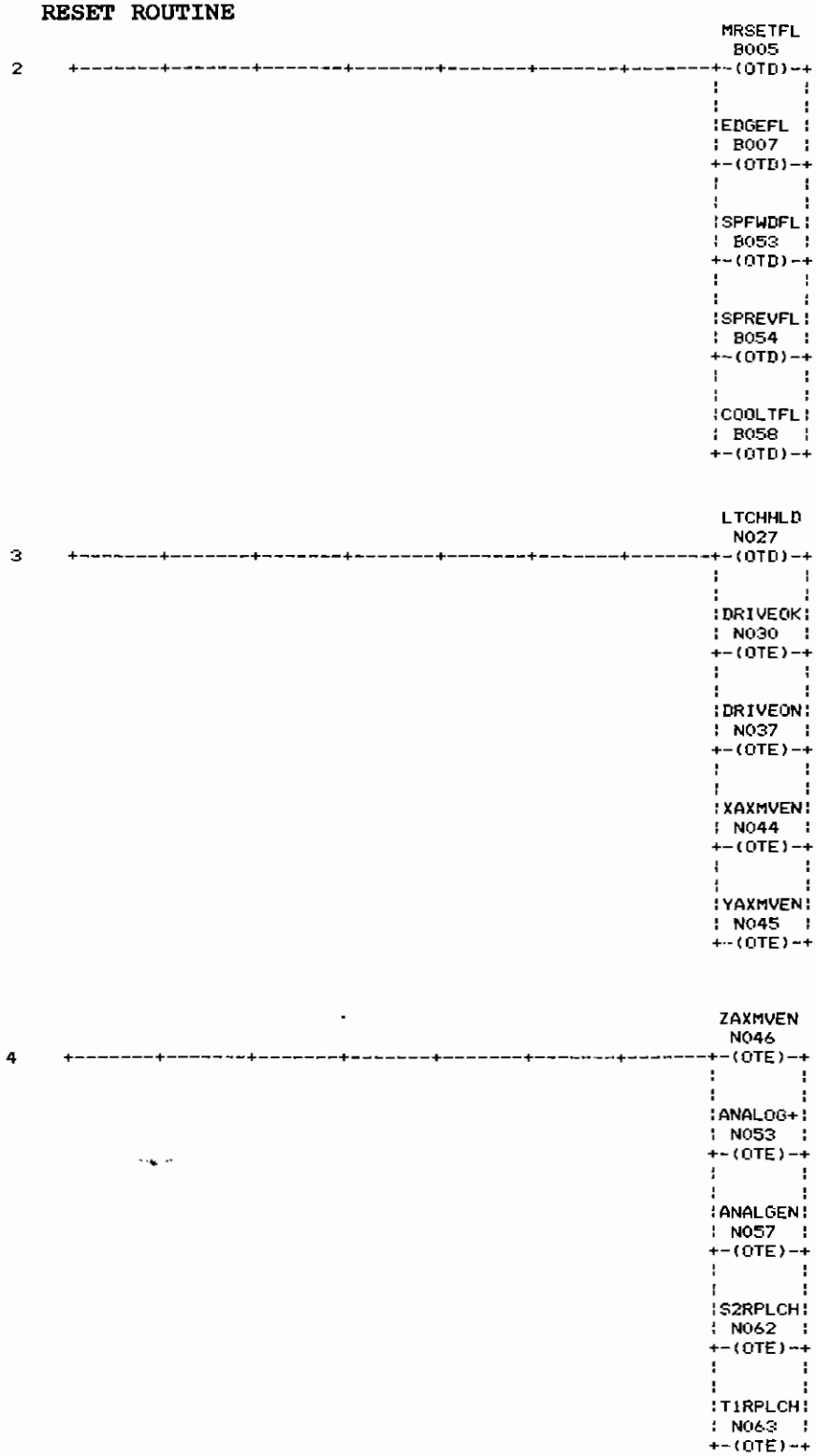
```
12-30-83  LADDER LIST: 8400 EXAMPLE PAL PROGRAM                PAGE    2

                          ALARM MESSAGES TEXT
M001 - REVERSE VIDEO = F BLINK = F     EMERGENCY STOP
M002 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 002
M003 - REVERSE VIDEO = F BLINK = F     AXIS DRIVE FAULT
M004 - REVERSE VIDEO = F BLINK = F     UNDEFINED M-CODE
M005 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 005
M006 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 006
M007 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 007
M008 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 008
M009 - REVERSE VIDEO = F BLINK = F     ALARM MESSAGE 009

                          STATUS MESSAGES TEXT
M010 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 001
M011 - REVERSE VIDEO = F BLINK = F     FLOOD COOLANT
M012 - REVERSE VIDEO = F BLINK = F     SPINDLE FORWARD
M013 - REVERSE VIDEO = F BLINK = F     SPINDLE REVERSE
M014 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 005
M015 - REVERSE VIDEO = F BLINK = F     AWAITING M-RESET
M016 - REVERSE VIDEO = F BLINK = F     DELAY
M017 - REVERSE VIDEO = T BLINK = T     LOW LUBE LEVEL
M018 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 009
M019 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 010
M020 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 011
M021 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 012
M022 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 013
M023 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 014
M024 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 015
M025 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 016
M026 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 017
M027 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 018
M028 - REVERSE VIDEO = F BLINK = F     PAL PROG. EXAMPLE
M029 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 020
M030 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 021
M031 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 022
M032 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 023
M033 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 024
M034 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 025
M035 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 026
M036 - REVERSE VIDEO = F BLINK = F     STATUS MESSAGE 027

                        VARIABLE MESSAGES TEXT
M037 - REVERSE VIDEO = F BLINK = F     VAL MSG 1
M038 - REVERSE VIDEO = F BLINK = F     VAL MSG 2
M039 - REVERSE VIDEO = F BLINK = F        RPM =
M040 - REVERSE VIDEO = F BLINK = F     VAL MSG 4
```

ENTER LADDER



If either Fault Clear or E-Stop is true, then do resets. If neither is true, then jump to the input detect routine.

RESET ROUTINE



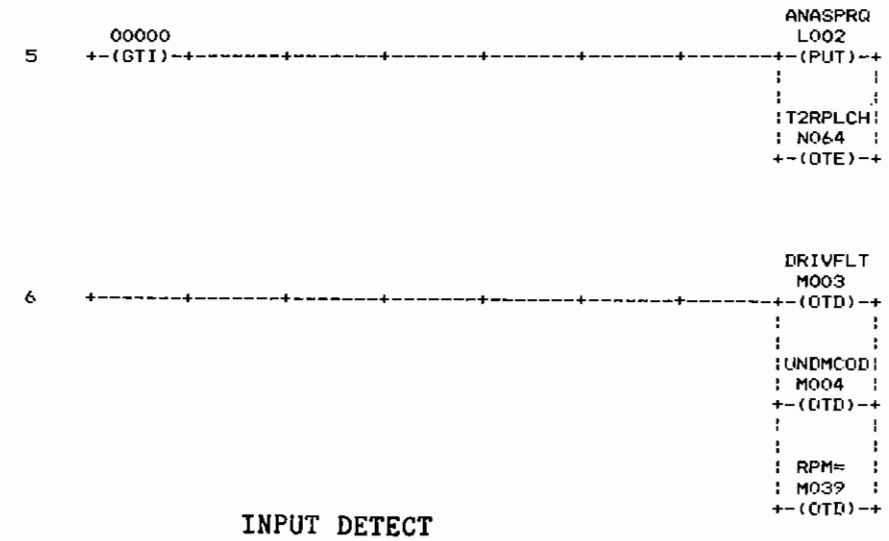
Reset B flags used in the PAL program (all B flags are reset on power-up).

Reset the latched feedhold flag, set the axis drives OK, set axis drives on, set X axis move enable and Y axis move enable.

The CNC will not allow axis motion unless the move enable flags are set (clamps off). These flags are usually used in conjunction with axis clamps.

Set Z axis move enable, set analog polarity positive, set analog output enable, set S2 reply check, set T1 reply check.

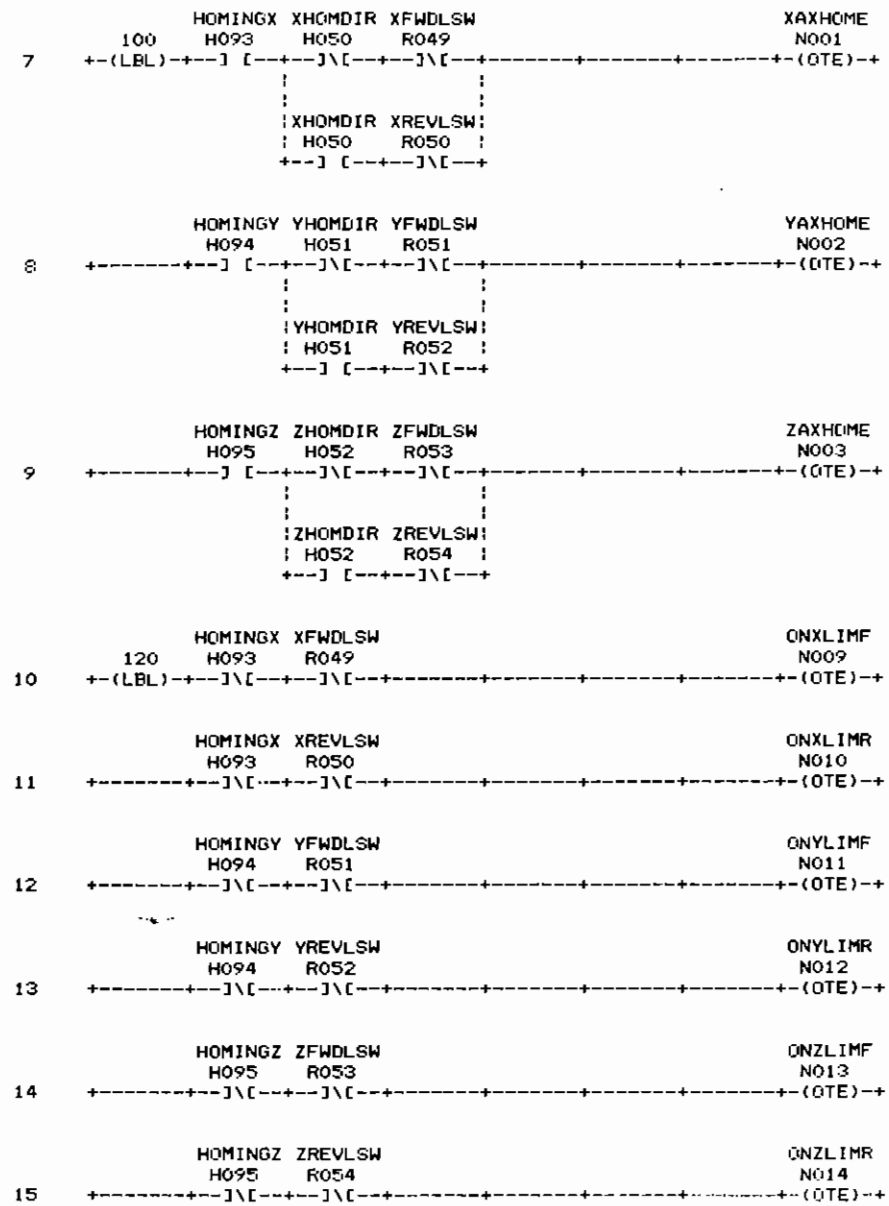
The S1-strobe is used for spindle speed and setting N062 assures that the CNC will not check for an S2 reply and hang up when the PAL program does not give an S2 reply for the S2 strobe. The T1 reply check is set for the same reason.



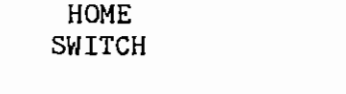
Put 0 in the analog speed register (causes analog output to be zero volts). Set T2 reply check, see above.

Reset any messages that are latched on.

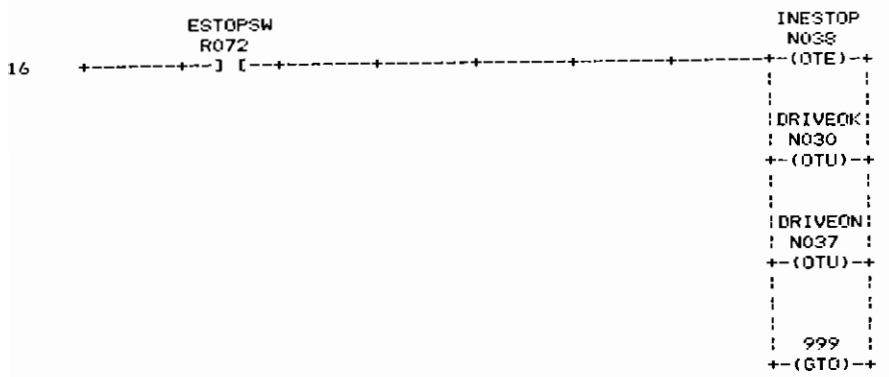
INPUT DETECT



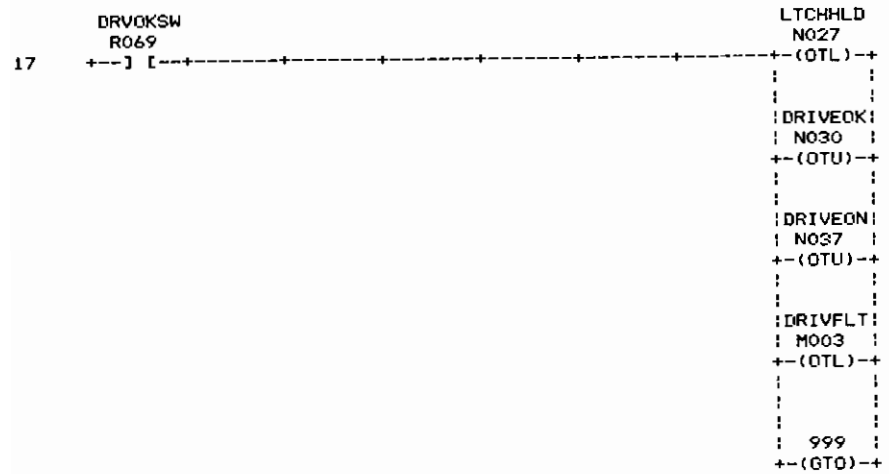
If axis homing flag is true, and homing direction flag is reset, then use forward travel limit switch for homing. If homing direction flag is set, use reverse travel limit switch. If separate home switches are used, the rung could be:



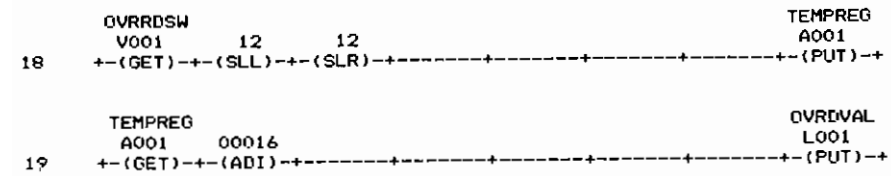
If axis homing flag is not set, then use the travel limit switches for travel limits. If separate home switches are used, then the axis homing flag is not used in these rungs.



When E-Stop switch is closed, set E-Stop flag, reset drive flags, and jump to the end of the PAL program.



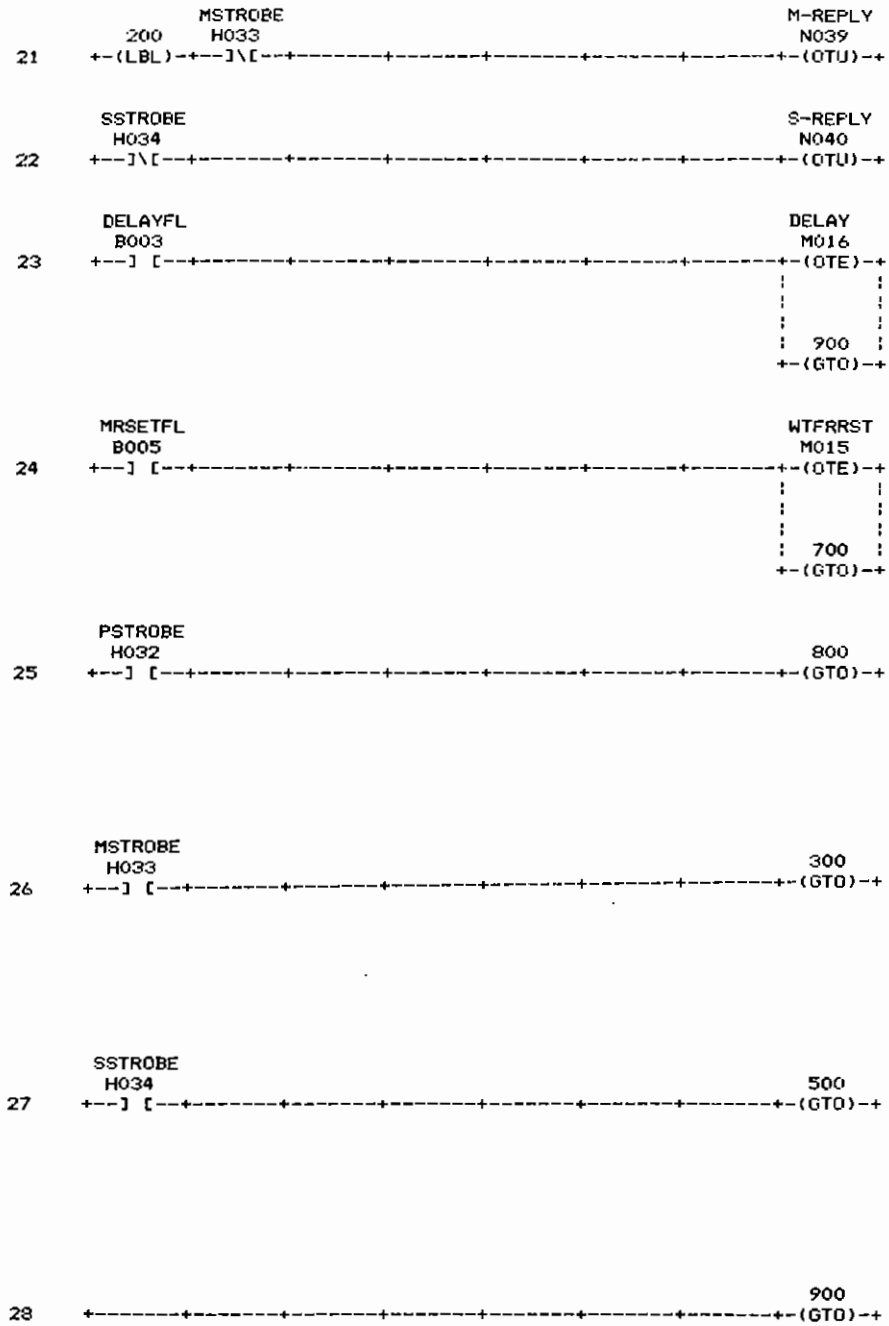
If a drive fault is detected, set the latched feedhold flags, reset the drive flags, set message, and jump to the end of the PAL program.



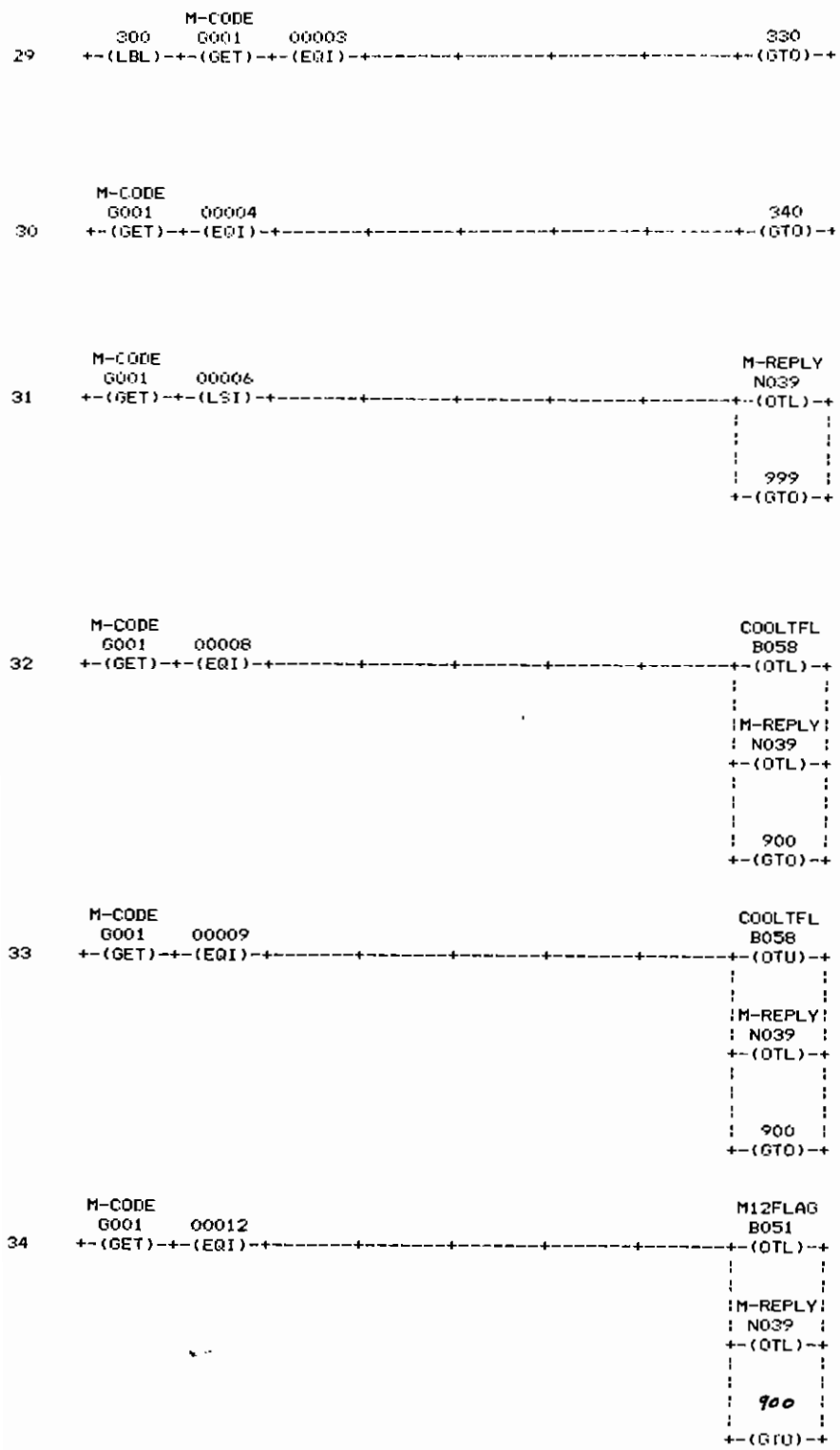
Get the port with the override switch as a V variable and put it into an A variable. For 8400 MP with non-sequential switches, it is necessary to add 16 before sending the value to the CNC. Sequential switches do not require this.

If the low lube switch is true, put message on the screen, but do not stop system operation.

ACTION DECODE



M-FUNCTION DECODE



If M3, then jump to spindle on forward routine.

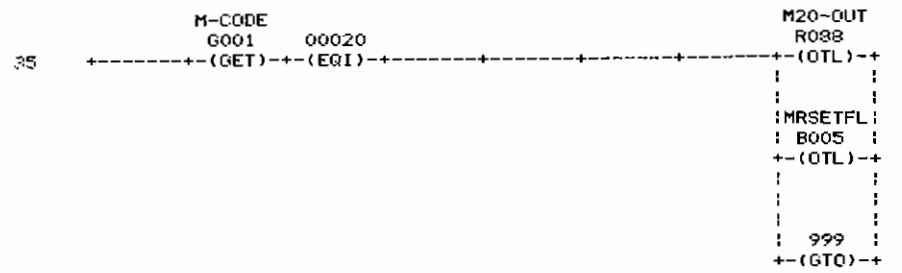
If M4, then jump to spindle on reverse routine.

If not M3 or M4, and M is less than 7, then M-function is post-move, so set the pre-move reply flag and jump to the end of the PAL program.

If M8, set the flood coolant flag, the M-reply flag, and jump to the output handling routine.

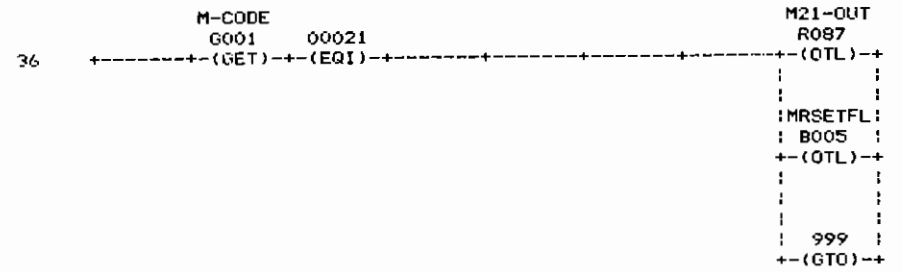
If M9, reset the flood coolant flag, and jump to the output handling routine.

If M12, set the M12 flag, M-reply flag, and jump to the output handling routine.



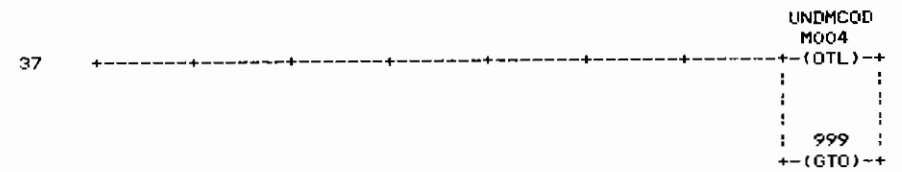
35

If M20, then set the M20 output, the wait for reset flag, and jump to the end of the PAL program.



36

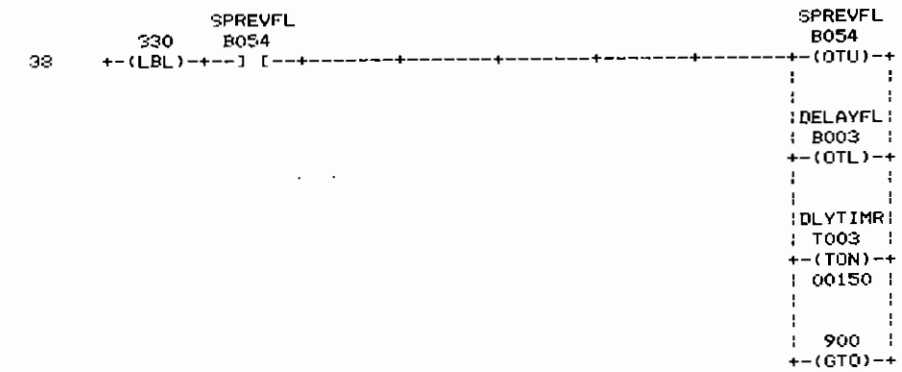
If M21, set the M21 output.



37

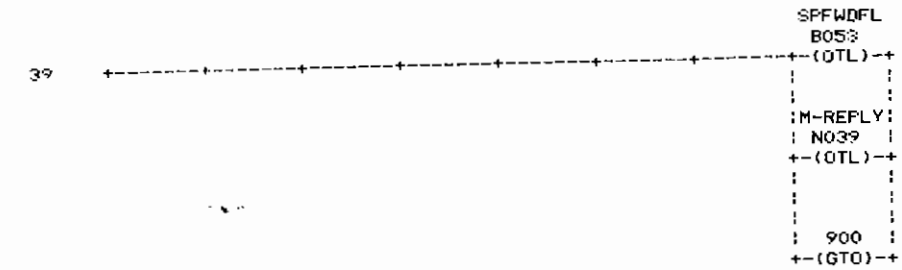
If the program reaches this point, then an undefined M-code has been programmed. Set alarm message and jump to the end of the PAL program.

SPINDLE ON FORWARD ROUTINE



38

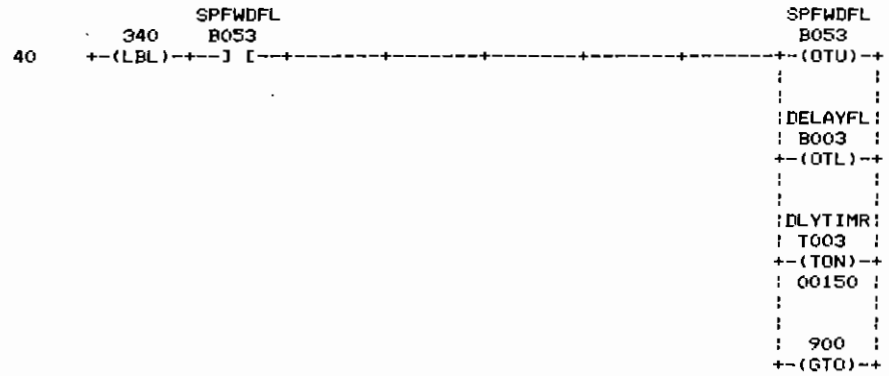
If the spindle is on reverse, then turn it off, set the delay flag, start the delay timer and jump to the output handling routine.



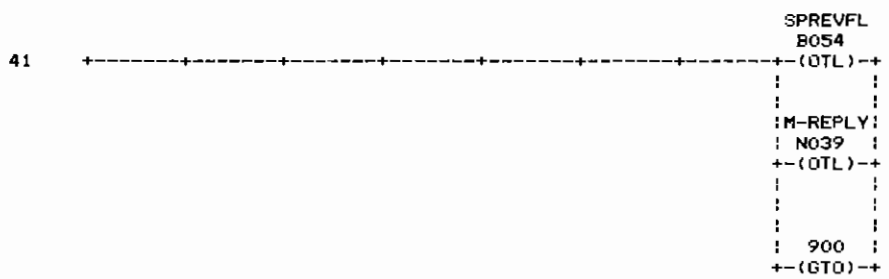
39

If spindle is off, or on forward, set the forward flag, the M-reply flag and jump to the output handling routine.

SPINDLE ON
REVERSE ROUTINE

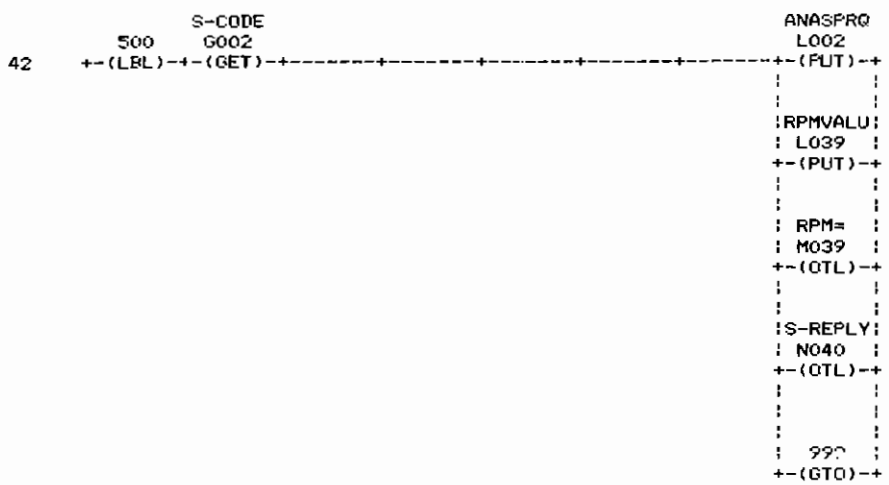


If the spindle is on forward, then turn it off, set the delay flag, start the delay timer and jump to the output handling routine.



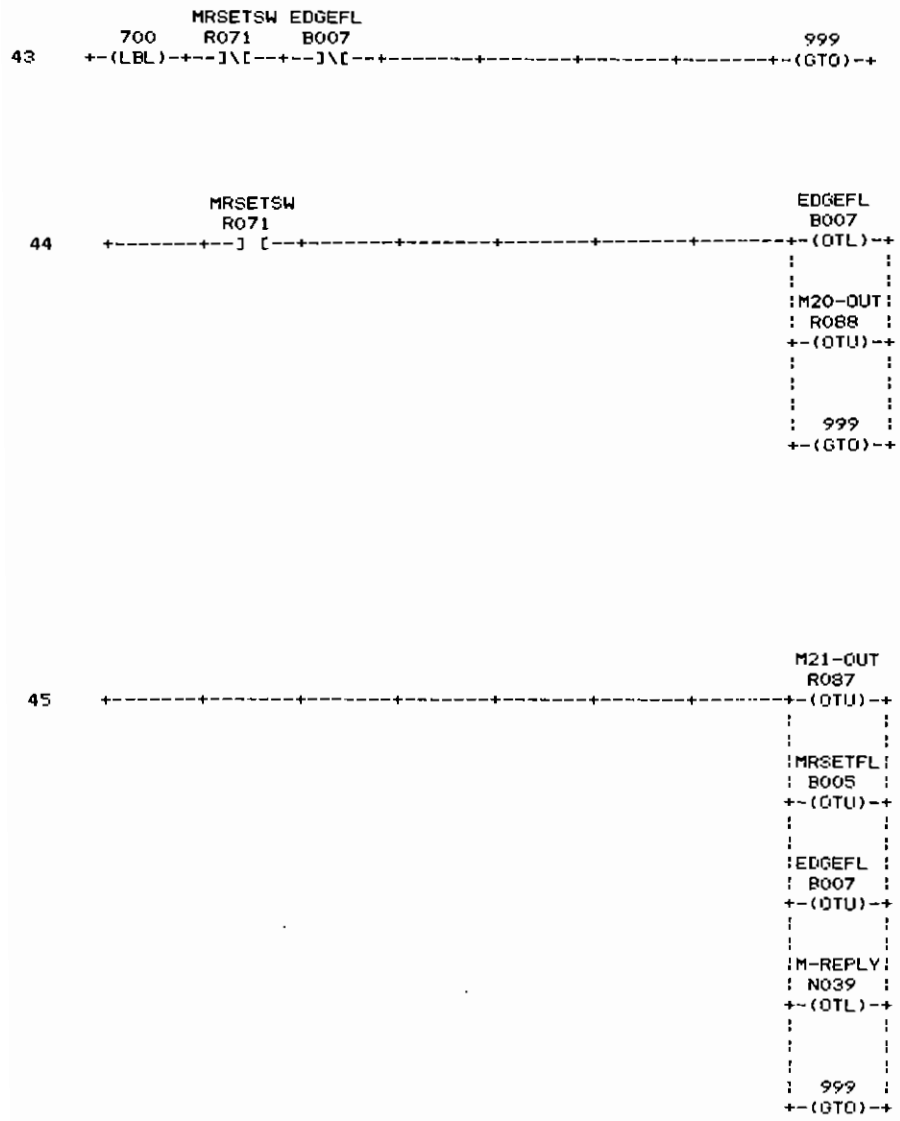
If spindle is off, or on reverse, set the reverse flag, the M-reply flag and jump to the output handling routine.

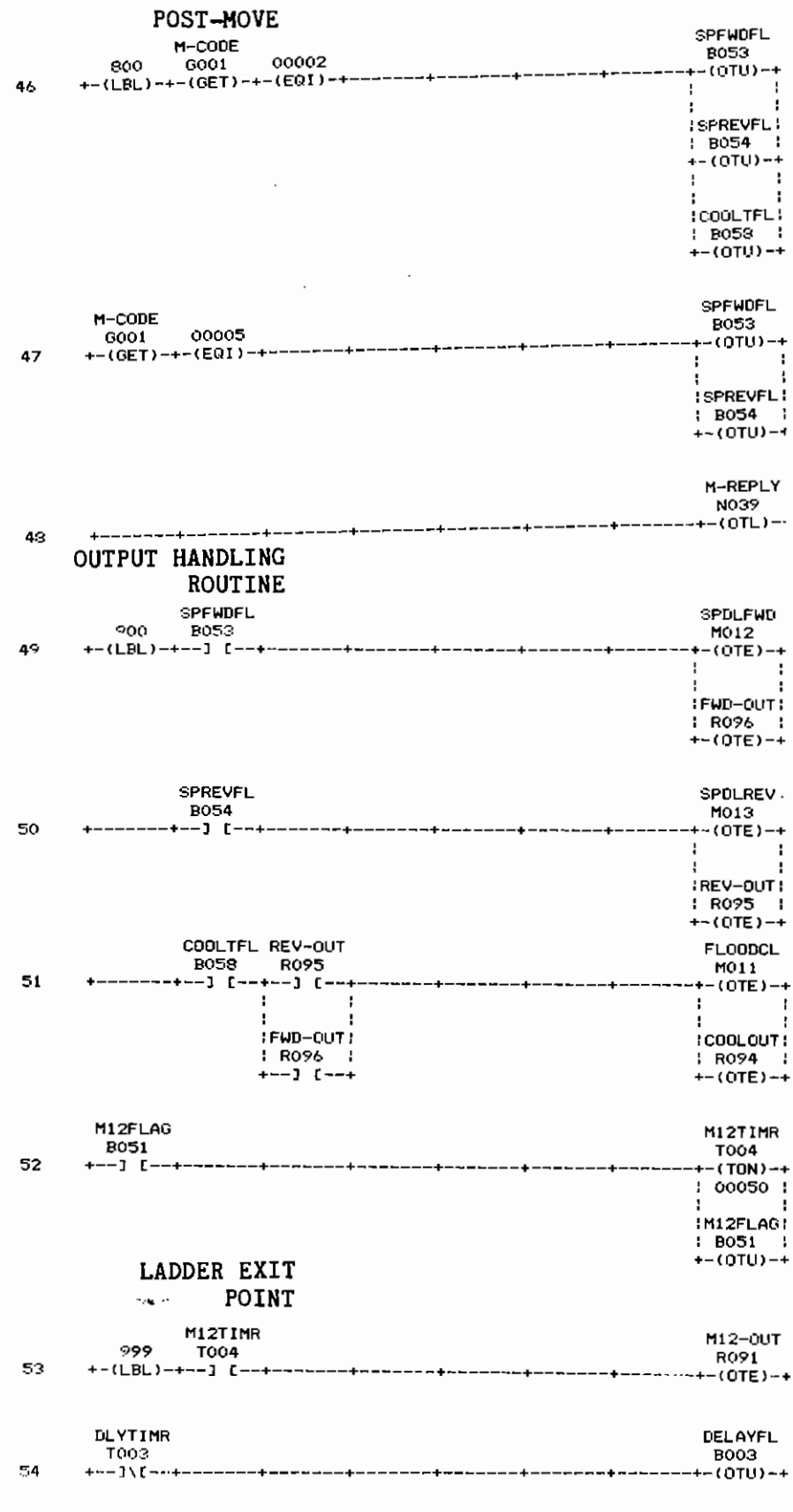
S-FUNCTION ROUTINE



Get S-code, put it in the spindle speed variable, the speed value variable, and enable the speed display message. Set the S-reply flag and jump to the end of the PAL program.

M-RESET DETECT
ROUTINE




POST-MOVE
M-CODE
800 G001 00002
46 +-(LBL)-+-(GET)-+-(EQI)-+
SPFWDFL
B053
+-(OTU)-+
SPREVFL
B054
+-(OTU)-+
COOLTFL
B058
+-(OTU)-+
On M2, turn off spindle and coolant.
M-CODE
G001 00005
47 +-(GET)-+-(EQI)-+
SPFWDFL
B053
+-(OTU)-+
SPREVFL
B054
+-(OTU)-+
On M5, turn off spindle.
M-REPLY
N039
48 +-(OTL)-
Set M-reply.
OUTPUT HANDLING ROUTINE
SPFWDFL
900 B053
49 +-(LBL)-+--] [-+
SPDLFWD
M012
+-(OTE)-+
FWD-OUT
R096
+-(OTE)-+
If the spindle forward flag is set, turn on message and the spindle on forward.
SPREVFL
B054
50 +--] [-+
SPDLREV
M013
+-(OTE)-+
REV-OUT
R095
+-(OTE)-+
If the spindle reverse flag is set, turn on message and the spindle on reverse.
COOLTFL REV-OUT
B058 R095
51 +--] [--+--] [-+
FWD-OUT
R096
+--] [--+
FLOODCL
M011
+-(OTE)-+
COOLOUT
R094
+-(OTE)-+
If the coolant flag is set and the spindle is on, then turn on the coolant and the coolant message.
M12FLAG
B051
52 +--] [-+
M12TIMR
T004
+-(TON)-+
00050
M12FLAG
B051
+-(OTU)-+
If the M12 flag is set, turn on the M12 timer for a half second and unlatch the M12 flag.
LADDER EXIT POINT
M12TIMR
999 T004
53 +-(LBL)-+--] [-+
M12-OUT
R091
+-(OTE)-+
If the M12 timer is on, turn on the M12 output.
DLYTIMR
T003
54 +--]\[-+
DELAYFL
B003
+-(OTU)-+
If the delay timer is not on, then reset the delay flag.
END OF PAL PROGRAM

This is the usage of immediate values in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
12	18	(SLL)	19	(SLR)				
100	1	(GTO)	7	(LBL)				
120	10	(LBL)						
200	21	(LBL)						
300	26	(GTO)	29	(LBL)				
330	29	(GTO)	38	(LBL)				
340	30	(GTO)	40	(LBL)				
500	27	(GTO)	42	(LBL)				
700	24	(GTO)	43	(LBL)				
800	25	(GTO)	46	(LBL)				
900	23	(GTO)	28	(GTO)	32	(GTO)	33	(GTO)
	38	(GTO)	39	(GTO)	40	(GTO)	41	(GTO)
	49	(LBL)						
999	16	(GTO)	17	(GTO)	31	(GTO)	34	(GTO)
	35	(GTO)	36	(GTO)	37	(GTO)	42	(GTO)
	43	(GTO)	44	(GTO)	45	(GTO)	53	(LBL)
00000	5	(GTI)						
00002	46	(EQI)						
00003	29	(EQI)						
00004	30	(EQI)						
00005	47	(EQI)						
00006	31	(LSI)						
00008	32	(EQI)						
00009	33	(EQI)						
00012	34	(EQI)						
00016	19	(ADI)						
00020	35	(EQI)						
00021	36	(EQI)						
00050	52	(TON)						
00150	38	(TON)	40	(TON)				

This is the usage of the A variables in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
A001	18	(PUT)	19	(GET)				

This is the usage of B flags in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
B003	23	-] [-	38	(OTL)	40	(OTL)	54	(OTU)
B005	2	(OTD)	24	-] [-	35	(OTL)	36	(OTL)
	45	(OTU)						
B007	2	(OTD)	43	-]\[-	44	(OTL)	45	(OTU)
B051	34	(OTL)	52	(OTU)	52	-] [-		
B053	2	(OTD)	39	(OTL)	40	(OTU)	40	-] [-
	46	(OTU)	47	(OTU)	49	-] [-		
B054	2	(OTD)	38	(OTU)	38	-] [-	41	(OTL)
	46	(OTU)	47	(OTU)	50	-] [-		
B058	2	(OTD)	32	(OTL)	33	(OTU)	46	(OTU)
	51	-] [-						

This is the usage of G variables in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
G001	29	(GET)	30	(GET)	31	(GET)	32	(GET)
	33	(GET)	34	(GET)	35	(GET)	36	(GET)
	46	(GET)	47	(GET)				
G002	42	(GET)						

This is the usage of H flags in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
H032	25	-] [-						
H033	21	-]\[-	26	-] [-				
H034	22	-]\[-	27	-] [-				
H044	1	-]\[-						
H050	7	-] [-	7	-]\[-				
H051	8	-] [-	8	-]\[-				
H052	9	-] [-	9	-]\[-				
H093	7	-] [-	10	-]\[-	11	-]\[-		
H094	8	-] [-	12	-]\[-	13	-]\[-		
H095	9	-] [-	14	-]\[-	15	-]\[-		

This is the usage of L variables in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
L001	19	(PUT)						
L002	5	(PUT)	42	(PUT)				
L039	42	(PUT)						

This is the usage of M flags in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
M003	6	(OTD)	17	(OTL)				
M004	6	(OTD)	37	(OTL)				
M011	51	(OTE)						
M012	49	(OTE)						
M013	50	(OTE)						
M015	24	(OTE)						
M016	23	(OTE)						
M017	6	(OTD)	20	(OTE)				
M039	6	(OTD)	42	(OTL)				

This is the usage of N flags in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
N001	7	(OTE)						
N002	8	(OTE)						
N003	9	(OTE)						
N009	10	(OTE)						
N010	11	(OTE)						
N011	12	(OTE)						
N012	13	(OTE)						
N013	14	(OTE)						
N014	15	(OTE)						
N027	3	(OTD)	17	(OTL)				
N030	3	(OTE)	16	(OTU)	17	(OTU)		
N037	3	(OTE)	16	(OTU)	17	(OTU)		
N038	16	(OTE)						
N039	21	(OTU)	31	(OTL)	32	(OTL)	33	(OTL)
	34	(OTL)	39	(OTL)	41	(OTL)	45	(OTL)
	48	(OTL)						
N040	22	(OTU)	42	(OTL)				
N044	3	(OTE)						
N045	3	(OTE)						
N046	4	(OTE)						
N053	4	(OTE)						
N057	4	(OTE)						
N062	4	(OTE)						
N063	4	(OTE)						
N064	5	(OTE)						

This is the usage of R flags in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
R049	7	-]\[-	10	-]\[-				
R050	7	-]\[-	11	-]\[-				
R051	8	-]\[-	12	-]\[-				
R052	8	-]\[-	13	-]\[-				
R053	9	-]\[-	14	-]\[-				
R054	9	-]\[-	15	-]\[-				
R069	17	-] [-						
R070	20	-] [-						
R071	43	-]\[-	44	-] [-				
R072	1	-]\[-	16	-] [-				
R087	36	(OTL)	45	(OTU)				
R088	35	(OTL)	44	(OTU)				
R091	53	(OTE)						
R094	51	(OTE)						
R095	50	(OTE)	51	-] [-				
R096	49	(OTE)	51	-] [-				

This is the usage of timers in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
T003	38	(TON)	40	(TON)	54	-]\[-		
T004	52	(TON)	53	-] [-				

This is the usage of V variables in the ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
V001	18	(GET)						

This is the cross reference of the mnemonics use in the example ladder.

VARIABLE	REFERENCED BY: RUNG	USAGE	RUNG	USAGE	RUNG	USAGE	RUNG	USAGE
RPM=	6	(OTD)	42	(OTL)				
ANASPRQ	5	(PUT)	42	(PUT)				
COOLOUT	51	(OTE)						
COOLTFL	2	(OTD)	32	(OTL)	33	(OTU)	46	(OTU)
	51	-] [-						
DELAY	23	(OTE)						
DELAYFL	23	-] [-	38	(OTL)	40	(OTL)	54	(OTU)
DLYTIMR	38	(TON)	40	(TON)	54	-]\[-		
DRIVEOK	3	(OTE)	16	(OTU)	17	(OTU)		
DRIVEON	3	(OTE)	16	(OTU)	17	(OTU)		
DRIVFLT	6	(OTD)	17	(OTL)				
DRVOKSW	17	-] [-						
EDGEFL	2	(OTD)	43	-]\[-	44	(OTL)	45	(OTU)
ESTOPSW	1	-]\[-	16	-] [-				
FLOODCL	51	(OTE)						
FLTCLR	1	-]\[-						
FWD-OUT	49	(OTE)	51	-] [-				
HOMINGX	7	-] [-	10	-]\[-	11	-]\[-		
HOMINGY	8	-] [-	12	-]\[-	13	-]\[-		
HOMINGZ	9	-] [-	14	-]\[-	15	-]\[-		
INESTOP	16	(OTE)						
LTCHHLD	3	(OTD)	17	(OTL)				
LUBELSW	20	-] [-						
LUBLEVL	6	(OTD)	20	(OTE)				
M-CODE	29	(GET)	30	(GET)	31	(GET)	32	(GET)
	33	(GET)	34	(GET)	35	(GET)	36	(GET)
	46	(GET)	47	(GET)				
M-REPLY	21	(OTU)	31	(OTL)	32	(OTL)	33	(OTL)
	34	(OTL)	39	(OTL)	41	(OTL)	45	(OTL)
	48	(OTL)						
M12-OUT	53	(OTE)						
M12FLAG	34	(OTL)	52	(OTU)	52	-] [-		
M12TIMR	52	(TON)	53	-] [-				
M20-OUT	35	(OTL)	44	(OTU)				
M21-OUT	36	(OTL)	45	(OTU)				
MRSETFL	2	(OTD)	24	-] [-	35	(OTL)	36	(OTL)
	45	(OTU)						
MRSETSW	43	-]\[-	44	-] [-				
MSTROBE	21	-]\[-	26	-] [-				
ONXLIMF	10	(OTE)						
ONXLIMR	11	(OTE)						
ONYLIMF	12	(OTE)						
ONYLIMR	13	(OTE)		ONZLIMR	15	(OTE)		
ONZLIMF	14	(OTE)						

This is the cross reference of the mnemonics use in the example ladder (continued).

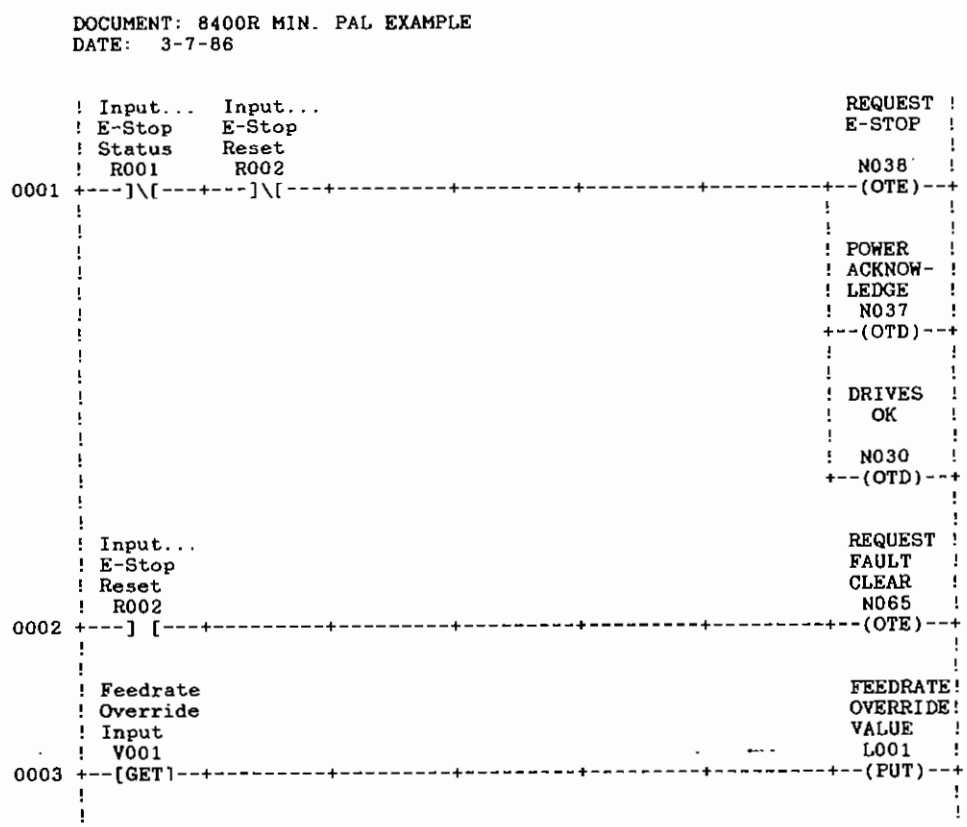
OVRDVAL	19	(PUT)								
OVRRDSW	18	(GET)								
PSTROBE	25	-] [-								
REV-OUT	50	(OTE)	51	-] [-						
RPMVALU	42	(PUT)								
S-CODE	42	(GET)								
S-REPLY	22	(OTU)	42	(OTL)						
S2RPLCH	4	(OTE)								
SPDLFWD	49	(OTE)								
SPDLREV	50	(OTE)								
SPFWDFL	2	(OTD)	39	(OTL)	40	(OTU)	40	-] [-		
	46	(OTU)	47	(OTU)	49	-] [-				
SPREVFL	2	(OTD)	38	(OTU)	38	-] [-	41	(OTL)		
	46	(OTU)	47	(OTU)	50	-] [-				
SSTROBE	22	-]\[-	27	-] [-						
T1RPLCH	4	(OTE)								
T2RPLCH	5	(OTE)								
TEMPREG	18	(PUT)	19	(GET)						
UNDMCOD	6	(OTD)	37	(OTL)						
WTFRRST	24	(OTE)								
XAXHOME	7	(OTE)								
XAXMVEN	3	(OTE)								
XFWDLSW	7	-]\[-	10	-]\[-						
XHOMDIR	7	-] [	7	-]\[-						
XREVLSW	7	-]\[-	11	-]\[-						
YAXHOME	8	(OTE)								
YAXMVEN	3	(OTE)								
YFWDLSW	8	-]\[-	12	-]\[-						
YHOMDIR	8	-] [-	8	-]\[-						
YREVLSW	8	-]\[-	13	-]\[-						
ZAXHOME	9	(OTE)								
ZAXMVEN	4	(OTE)								
ZFWDLSW	9	-]\[-	14	-]\[-						
ZHOMDIR	9	-] [-	9	-]\[-						
ZREVLSW	9	-]\[-	15	-]\[-						

The following example is the minimum start up ladder for the 8400 Series Robot control. This program is intended for your information only. In no way does this represent a fully functional ladder.

We recommend that you use this ladder with the ladders in Chapter 3 on page 21 and 22, and the ladder in Appendix F to practice using the Kaypro, and to study the concepts involved in PAL development.

The list provided with the advanced example in Appendix F on page F-1 was generated using a development system that is more sophisticated than the one provided with the Kaypro. This is why you may notice differences between what is given, and what you write on the Kaypro.

Example Ladder for Series 8400 Robot Control



Example Ladder for Series 8400 Robot Control (continued)

DOCUMENT: 8400R MIN. PAL EXAMPLE
DATE: 3-7-86

PAGE: 15