

Centurion IV

Programming Manual

Version 1.07

June 1, 1986

Milltronics Manufacturing Company

1400 Mill Lane
Waconia, MN 55387
(612) 442-1401

SUBJECT INDEX

	<u>PAGE</u>
<u>I. MACHINE MOVEMENT</u>	
1. Auto Grid Zero 1-11A, 1-11B, 2-3, 2-12, 3-1, 4-1, 18-14	1-1
2. Axis Movement MDI	4-5
3. Handwheel 2-1, 2-2, 4-4, 6-2, 18-15	1-5
4. Jog 1-11A, 2-1, 2-4, 4-2, 4-2a, 4-3, 18-5	1-1
<u>II. SET FLOATING ZERO</u>	1-1
1-9, 2-37, 4-1, 4-2, 4-2a, 18-14	
<u>III. RS-232, MAGNETIC AND PAPER TAPE OPERATION</u>	
1. Format Magnetic Tape	16-3
2. Write Magnetic Tape 5-11, 5-12	2-5
3. Read Magnetic Tape 5-11, 5-12	2-5
4. Write System Parameter Tape	15-4
5. Read System Parameter Tape	15-4
6. Read System Parameter From Paper Tape	15-4
7. Set RS-232 Baud Rate 5-18	15-3A
8. Dump System Parameter to RS-232	15-14
9. Read From RS-232 to Memory 2-29, 5-17	1-9
10. Dump Memory to RS-232 2-29, 5-17	1-9
11. Read Paper Tape 1-9, 2-20A, 2-29, 5-13, 5-14, 5-15, 5-16, 5-17, 15-3A	1-8
<u>IV. SETTING TOOL OFFSETS AND TOOL CHANGE</u>	
1. Tool Change 2-37, 17-1 thru 17-23	2-19
2. Tool Length Offset 6-1 thru 6-4	18-14
3. Modifying Tool Offsets In The Program thru 6-4	6-1
4. Modify Tool Offsets Via MDI thru 6-4	6-1
5. Activate Tool Offsets In Program	2-37
<u>V. G-CODES</u>	
1. List of G-Codes 2-8, 2-9	1-9
2. Detailed Description of G-Codes thru 2-39	2-20D

	<u>PAGE</u>
VI. <u>M-CODES</u>	
1. List Of M-Codes	1-8
2-10	
2. Detailed Description of M-Codes	2-19
thru 2-20C	
VII. <u>ENTER AND DISPLAY AND PROGRAMS</u>	
1. Enter Programs	5-1
thru 5-10, 18-1 thru 18-14	
2. Run Programs	7-1
7-2, 18-4, 18-14, 18-15	
3. Enter Programs Via Help	18-1
thru 18-4, 18-9, 18-10, 18-11	
VIII. <u>CIRCULAR AUTOROUTINES AND ARCS</u>	
1. Spiral Pocket Clear	2-27
2-38, 9-9 thru 9-11	
2. Arcs	1-2
2-21 thru 2-25, 2-27, 7-6 thru 7-8, 9-1 thru 9-4, 9-9	
thru 9-11, 11-1 thru 12-3, 13-11 thru 13-14, 17-1 thru 17-27	
3. Circular Finish	2-38
9-1, 9-2, 9-3, 9-4	
IX. <u>RECTANGULAR AUTOROUTINE</u>	
1. Rectangular Pocket Clear	2-26
2-38, 9-7, 9-8, 9-9	
2. Rectangular Pocket Finish	2-25
2-38, 9-3, 9-4, 9-5, 9-6, 9-7	
X. <u>DRILLING</u>	
1. Drill Can Circle	2-32
2-38, 7-4, 8-1, 8-2, 9-11, 17-15 thru 17-19, 18-10, 18-11	
2. Drill Dwell Can Cycle	2-32
2-38, 8-1, 9-11	
3. Drill Peck Can Cycle	2-33
2-38, 8-1, 9-11	
4. Wood Peck Drill Can Cycle	2-36
2-38, 8-1, 9-11	
5. Bolt Cycle Circles	1-10
2-38, 9-11, 9-12, 9-13	
XI. <u>BORING AND TAPPING</u>	
1. Bore	1-9
2-35, 2-38, 8-1	
2. Bore Dwell	1-9
2-36, 2-38, 8-1	
3. Tap	1-9
2-34, 2-38, 8-1	

	<u>PAGE</u>
XII. <u>MILLING</u>	
1. Milling and Milling Cycles	1-1
2-20B, 2-20C, 2-21 thru 2-31, 2-35, 2-36, 7-2 thru 7-8, 9-1	thru 9-11, 11-1 thru 11-3, 12-1 thru 12-3, 18-9
2. Milling Sample Programs	17-1
thru 17-27	
XIII. <u>SPECIAL PROGRAM FUNCTIONS ROTATION, SCALING, MIRROR IMAGE, ROUNDDING CORNERS, GENERATE CIRCLE 3 POINTS.</u>	
1. Round Corners (RNDC)	18-5
18-6	
2. 3 Point Circle Generate (CGEN)	18-6
18-7	
3. Scaling	1-7
2-38, 19-2, 19-3	
4. Rotation	1-7
2-38, 19-3	
5. Mirror Image	1-1
2-29 thru 2-31, 17-13, 17-14, 17-14a	
XIV. <u>POLAR CO-ORDINATE PROGRAMMING</u>	
1. Polar Lines	19-1
2. Polar Arc and Circles	19-2
XV. <u>CUTTER RADIUS COMPENSATION</u>	
1. G41, G42	1-3
1-9, 2-20A, 2-20B, 2-28, 9-3, 9-5, 9-6 thru 9-11, 13-1 thru	
13-14, 17-1 thru 17-27, 18-7, 18-12, 18-13	
XVI. <u>SUBROUTINES AND LOOPS</u>	
1. Subroutines	1-4
1-10, 2-38, 10-1 thru 10-4, 17-15 thru 17-19	
XVII. <u>PARAMETRIC PROGRAMMING</u>	14-1A
thru 14-4	
XVIII. <u>TRIG-HELP</u>	13-11
thru 13-14	
XIV. <u>HELP PROGRAMMING</u>	18-1
thru 18-15	
XX. <u>USE OF PARAMETRIC SUBROUTINES</u>	20-1
thru 20-30	

INDEX

<u>SECTION</u>		<u>PAGE</u>
I	<u>CONTROL FEATURES</u>	
1.1	Standard Features Available in all IIC Controls	1-1
1.1.1	Software Package Option I	1-6
1.1.2	Software Package Option II	1-7
1.2	M-Functions	1-8
1.3	G-Codes	1-9
1.4	Error Codes	1-11
1.5	Machine Commands	1-12
1.6	Axis Definitions	1-13
II	<u>OPERATORS CONTROL CONSOLE</u>	
2.1	Panel Layout	2-1
2.2	Control Console Switches	2-1
2.2.1	Pushbuttons and Selector Switches	2-1
2.2.2	Keypad Keys	2-2
2.2.3	Numeric Keypad	2-5
2.3	Display Status Features	2-6
2.3.1	Display Status Page 0 (control status)	2-6
2.3.2	Display Status Page 1 (G-Codes)	2-8
2.3.3	Display Status Page 2 (G-Codes)	2-9
2.3.4	Display Status Page 3 (M-Functions)	2-10
2.3.5	Display Status Page 5 (I/O Channels)	2-11
2.3.6	Display Status Page 6 (Tool Offsets)	2-13
2.3.7	Display Status Page 7 (Program Parameters)	2-17
2.3.8	Display Status Page 9 (System Parameters)	2-17
2.4	M-Functions Defined	2-19
2.5	Mutually Exclusive M-Codes	2-20
2.6	G-Codes Defined	2-20
2.7	Mutually Exclusive G-Codes	2-29
III	<u>MACHINE START-UP PROCEDURE</u>	
3.1	Machine Start-up After an Electrical Shutdown	3-1
3.2	Machine Start-up From Normal Shutdown	3-1
IV	<u>BASIC CONTROL OPERATION</u>	
4.1	Grid (Machine Reference)	4-1
4.2	Setting X, Y Floating Zero "0" (Part Reference)	4-1
4.2.1	Methods of Finding an X, Y Floating Zero	4-2
4.3	Jog Mode	4-2
4.3.1	Jog Pushbutton	4-3
4.3.2	Feed Selection	4-3
4.3.3	Incremental Feed Selection	4-3
4.4	Electronic Handwheel	4-4
4.5	MDI General	4-5
4.5.1	Do's and Don'ts of MDI Mode	4-6

<u>SECTION</u>		<u>PAGE</u>
V	<u>DISPLAY AND ENTER PROGRAMS</u>	
	5.1 Display and enter Programs	5-1
	5.2 Program Entry	5-1
	5.3 Program Display	5-4
	5.3.1 Do's and Don'ts of Program Entry	5-6
	5.4 Program Editing	5-6
	5.4.1 To Delete a Command From a Block	5-7
	5.4.2 To Change a Block Within A Program	5-8
	5.4.3 To Delete an Entire Block	5-8
	5.4.4 To Add Information to a Block of a Program	5-9
	5.4.5 To Add a Block Within a Program	5-10
	5.4.6 To Erase a Whole Program	5-10
	5.4.7 To Erase The Whole Memory	5-11
	5.5 Read and Write Tape Features	5-11
	5.5.1 Procedure Used to Write a Section of Magnetic Tape	5-11
	5.5.2 Procedure Used to Read a Section of Magnetic Tape into Memory	5-12
	5.5.3 Procedure for Editing Information Recorded on the Magnetic Tape	5-12
	5.5.4 Do's and Don'ts of Magnetic Tape Read/Write	5-13
	5.5.5 Procedure for Reading Paper Tape	5-13
	5.5.6 Paper Tape Standard	5-14
	5.5.7 Paper tape Format	5-15
	5.6 RS-232 and TTY Inputs/Outputs	5-17
VI	<u>TOOL OFFSETS</u>	
	6.1 Entering Tool Offsets	6-1
	6.2 Entering tool Length Offsets	6-1
	6.2.1 Key Sequence for Entering Radius, Fixture and Tool Length Offsets	6-2
	6.3 To Enter Tool Offsets from a Program or From the MDI Mode	6-3
	6.4 Modifying Tool Offsets via the MDI or Program Modes	6-4
VII	<u>RUNNING BASIC PROGRAMS</u>	
	7.1 Running Basic Programs	7-1
	7.1.1 Dry Run	7-1
	7.2 Codes Which are Activated When the Control is First Turned "On"	7-1
	7.3 Basic Set-up Prior to Running a Program	7-2
	7.4 Point to Point Programming	7-3
	7.5 Linear Interpolation or Angular Milling	7-5
	7.6 Circular Interpolation Programming	7-6
VIII	<u>USE OF CANNED CYCLES</u>	
	8.1 Turning Canned Cycles On and Off	8-1
	8.2 Examples Showing the Use of Some Canned Cycles	8-2

<u>SECTION</u>		<u>PAGE</u>
<u>IX</u>	<u>AUTOROUTINES</u>	
9.1	G26-Circular Finish	9-1
9.2	G26-Rectangular Finish with Circular or Square Corners	9-3
9.3	G27-Rectangular Pocket Clearing	9-7
9.4	G28-Circular Pocket Clearing with Finish Cut	9-9
9.5	Bolt Hole Circle Routine	9-11
<u>X</u>	<u>SUBROUTINES AND LOOPS</u>	
10.1	Program Entry of Subroutines	10-1
10.2	Using Subroutines as Fixture Offsets	10-1
10.3	Initiating and Looping Subroutines	10-1
<u>XI</u>	<u>SWITCHABLE PLANE PROGRAMMING</u>	
11.1	Programming Arcs in the XZ Plane	11-1
11.2	Programming Arcs in the YZ Plane	11-2
<u>XII</u>	<u>USING THE HELICAL PROGRAMMING FEATURE</u>	
12.1	What is Helical?	12-1
12.2	Helical Programming Examples	12-2
<u>XIII</u>	<u>CUTTER RADIUS COMPENSATION</u>	
13.1	Introduction	13-1
13.2	Sample Part Exercise	13-3
13.3	Tool Manipulation	13-7
13.3.1	Notes on Cutter Compensation	13-10
13.3.2	Final Note	13-10
13.4	"Trig-Help" Programming Aid Package	13-11
13.4.1	Things to Remember When Estimating Points	13-14
<u>XIV</u>	<u>FORMAT FOR PARAMETRICS</u>	14-1
<u>XV</u>	<u>SYSTEM PARAMETER SET-UP PAGE</u>	15-1
<u>XVI</u>	<u>DIAGNOSTICS</u>	
16.1	ROM Memory Diagnostcs	16-1
16.2	RAM Memory Test	16-2
16.3	Display Status Page 5 (Keypad Test)	16-3
16.4	Erase Magnetic Tape	16-3
<u>XVII</u>	<u>PROGRAMMING PROBLEMS</u>	
17.1	Angles Tangent to Arcs	17-1
17.2	Tangent Arcs	17-3
17.3	Tangent Arcs and Angles Tangent to Arcs	17-5
17.4	Non-Tangent Arcs (1)	17-6
17.5	Non-Tangent Arcs (2)	17-9
17.6	Non-Square Corner Start Position	17-10
17.7	Concentric Arcs Having Equal Radii	17-11
17.8	Firehydrant/Mirror Image	17-13
17.9	Hole Drilling and Tapping Pattern	17-15
17.10	Bolt Hole Pattern	17-16
17.11	Partial Bolt Hole Routine	17-18

<u>SECTION</u>		<u>PAGE</u>
	17.12 Faceplate	17-19
	17.13 Pocket Clearing ROUTINE FOR Circular Pockets of Varied Sizes	17-21
	17.14 Three Dimensional Contouring	17-22
	17.15 Student Program	17-24
	17.16 Student Program	17-26
XVIII	18.1 "HELP" Promp programming	18-1
XIX	19.0 Polar Programming & Part Rotation	19-1
XX	20 Use of Parametric Subroutines	20-1

MILLTRONICS

MANUFACTURING COMPANY

The Centurion IV by IIC, Inc. offers powerful features yet is easy to use. Take a moment and read through the Centurion IV CNC features and compare its capabilities to any other CNC on the market.

Interactive "Help"

The Centurion IV CNC can be programmed in two completely independent modes. The "Help" mode of programming leads the operator through a series of interactive, menu driven screens, that result in "blueprint" parts programming without knowledge of "M" and "G" codes. Often it is very easy to overlook important data needed in programming a part. The "Help" format will prompt the operator to give all the pertinent information before it will accept the next block. "Help" also establishes a simple method to get tool radius compensation turned on and off and approach and leave the part surfaces in an appropriate manner. The "Help" mode of input is an easy and comprehensive way to program a part. It is useable by anyone with minimal training. As an example, it will interactively set-up a mill cycle for working the tool down to a final depth, by repeating a programmed shape or cavity while incrementing Z down between each pass. The other mode of programming is the standard EIA N.C. programming language. The Centurion IV CNC is at home with the old standard CNC language and offers the future trend in programming CNC's in the form of the "Help" mode.

Events Operation

The "Help" mode of programming defines all positioning moves, cutting cycles, and tool change setups as "Events".

Auto Grid Zero

When selected, automatically sends all axes to a precise location. This feature simplifies power-up or set-up by offering a day-to-day precise reference point.

Part Rotation

This enables a parts program to be rotated in increments of .001 degrees. A program can be looped in a subroutine to result in incremental rotation each time.

Full Floating Zero

Floating zero can be selected at any desirable point. For setting a floating zero it does not matter where the machine is positioned. Axis can be preset to any desired position. This can be done either in a program or in MDI (Manual Data Input).

Absolute/Incremental Programming

On command, the CNC will convert to Incremental mode. Programs may switch back and forth between the Incremental and Absolute. Tool radius cutter compensation works for either format.

Inch/Metric Programming

On command, the CNC converts to programming in Metric rather than Inch.

Mirror Image

On command, the CNC will Mirror Image one or more axes. This is used to make "Left" and "Right" hand parts.

Scaling

This feature allows for scale factors between the range of +99.9999 to -99.9999 for each axis. Subroutines can be called repetitively with different scale factors.

Jog

Allows operator to Jog any axis or any combination of axes.

Electronic Handwheel*

Complements Jog and Incremental Jog, offering precise operator control of all axes by Electronic Handwheel. This is an invaluable feature for machine set-up. The position display operates as a digital readout system while handwheeling. This feature can be used to set floating zeroes, preset tool lengths on the machines and even generate a program while handwheeling from point to point. Electronic Handwheel resolution control allows the operator to select the sensitivity of the handwheel.

Incremental Jog

Allows operator to select Incremental moves of .0001", .001", .010" or .1" for any axis in any direction.

Feedrate Override

Modifies the programmed feedrate or jog feedrate from 0% to 120%. Also selects the electronic handwheel resolution.

Tool Length Offset Combination

Tool length offsets can be preset manually, in a program or with the Electronic Handwheel. Tools are simply handwheeled down to a reference surface and the "Enter" button pushed. Up to 24 tools can be stored in the tool table to match your requirements.

Fixture Offset Compensation

Enables X & Y axes to be shifted to enable use of two fixtures on the table.

Linear Interpolation

The control linearly interpolates up to four axes in any command Point to Point move. A constant surface feedrate is maintained.

Circular Interpolation

Simultaneous circular interpolation can be programmed for any two out of four axes. The radius can be up to 100 inches with .0001" resolution and can interpolate up to 100 inches per minute. A full circle can be programmed in one instruction. Circular programming is accomplished with either radius, center point (I, J) and end point (X,Y), or radius, start angle, end angle, or a combination of input information. The system also can accept arc descriptions which are missing the center point.

Helical Interpolation

This enables the programming of three axes of movement with two axes in circular interpolation and the third axes in linear interpolation.

Continuous Path

The system can operate in either Continuous Path (Corner Round Mode) or Stop Mode (Point-to-Point). The Corner Round mode eliminates dwell marks for a smooth mill finish. The Point-to-Point mode is used for cases such as drilling.

Can Cycles

Can Cycles are activated in "Help" mode through a Menu or by G-codes in Std. N.C. mode. Can cycles include Drilling, Peck Drilling, Partial Peck Drilling, Milling, Bore, Bore with Dwell and Tapping. Can cycles usually work from the reference R-plane and substantially reduce program length.

Automatic R-plane

An "R" (reference) plane can be established as a clearance plane just above the work piece. This "R" plane will be referenced to each tool called up. It greatly shortens programming when using Can Cycles.

Autoroutine

Autoroutines include various cycles for rectangular pocket clearing, rectangular pocket with radius corner finishing, bolt circle, partial bolt circle, circular clearing and circular finishing. Can cycles, subroutines and tool radius compensation can all be used with autoroutines.

Subroutines

Subroutines can be written up to 10 levels deep. They can be edited as easily as the main program. Any function which can be acted upon in the main program can be acted upon in the subroutines. Subroutine call-ups and exits will remain correct after line insert or line deletes have been performed. All subroutines set a virtual "floating zero" from their point of entry making the programming much simpler. Subroutines can also be looped up to 990 times.

True Decimal Point Programming

The CNC utilizes a calculator type keypad for fast easy entry. Decimal points can be initiated at any time while entering or will be automatically inserted after the correct number of digits.

Eight Program Storage

The CNC can store and recall up to eight separate programs. Any of the eight programs stored can call-up subroutines and each remains independent for editing.

Full Line/Event Edit

Any program line or part of a line can be easily deleted, edited, or expanded. In the "Help" mode Events can be easily edited.

Line/Event Insert and Delete

Editing would be very limited without the capability to insert or delete lines (Events in "Help"). The CNC allows you to insert as many lines as you want, wherever you want and will automatically resequence the program line numbers and retain proper subroutine call-up instructions.

Additional Memory*

Up to 30K of memory is available. The amount of unused memory can be displayed at any time.

Edit Key

To prevent undesired editing or erasing, an edit key is provided which locks out all ability for the operator to change the parts program memory.

Line Reorganization

If line instructions are entered in improper order, the CNC will automatically reorganize the information in the correct order.

Man Readables

Statements may be entered into the program to remind the operator of a task or make any form of description desired. Input of alphabetical characters can be done by paper tape, RS-232 port or keypad.

Run Program

Any of the eight programs can be called up to run and can be started at any valid line number. Before actual program execution begins, a check of the entire program is made by the CNC. Many types of errors are located by the CNC during this check. If the CNC locates a program error it will display the error type and the location of the error. This can save program verification time and prevent scrapping of parts.

Auto Block Switch

The operator can select automatic operation or single block operation.

Error Messages Displayed

The CNC presents many error messages to the operators. They may indicate illegal programming, syntax errors, loss of power, low level of machine lubrication and many more. Most errors are explained in plain English.

Excess Error Protection

The CNC continuously monitors the actual position versus the command position. Should the actual machine position lag excessively behind the commanded position, a servo shutdown takes place and the operator receives a message indicating that the machine is in excess error condition.

DC Closed Loop Servos

The CNC is designed to operate any closed loop servo system. Feedback can be derived either from a linear scale or an incremental rotary encoder. The DC analog is created in real time and updated more than 4000 times a second to maximize servo performance and acceleration/deceleration ramps. In addition, the servo loop locks into the feedback pulse leading edge thus eliminating "hunting" from pulse-to-pulse. This results in extremely good part finishes and maximizes servo life. Servo positioning resolution is 0.0001".

Electronic Software Limits

Before the CNC executes a machine motion block, the CNC checks to see if the move will result in an over travel condition. In case it would, the execution is interrupted and an appropriate message is displayed.

Full Program Look-Ahead Tool Radius Compensation

The CNC will automatically adjust the program path to compensate for a tool radius. The program can be compensated from the blueprint path to either the left or right. Any tool radius can be entered and no special program editing is necessary. The CNC will cutter radius compensate autoroutines, subroutines, all arc intersection, and line intersection without special consideration or insertions of extra lines of program. If the new tool path (radius compensated path) is illegal, i.e. tool too large for inside radius, the CNC will flag the operator with the type of error and the line number where the error occurs. The CNC will not run a program until it knows that the cutter compensation is valid throughout the entire program. This substantially helps the operator to verify his program for correct data, do whatever editing is necessary and prevent scrap parts.

Trig-Help

Along with the cutter compensation the Trig-Help feature greatly simplifies arc programming. The CNC will often locate intersection co-ordinate points in arc-to-arc moves, line-to-arc moves and arc-to-line moves. In arc-to-arc cases, the operator need only guess at the intersection point and the CNC will correct the errors. Where straight lines intersect arcs, the operator need only choose a point located on the straight line and not necessarily the intersection co-ordinate point. This feature eliminates the need to trig out many complex intersection points, saving programming time and making MDI (manual data input) input much easier. Trig-Help is not found in most CNC's, thus forcing the operator to calculate these points precisely. In many cases, the necessary calculations can be very difficult.

Auto Corner Rounding

This feature allows automatic rounding of corners made by intersecting lines and circles. The operator simply specifies the radius needed at the corner. This is a Tremendous time saver.

3 Point Circle Generate

This feature is very handy when 3 points on a circumference are known. The CNC can take these operator given points and substitute an appropriate arc that connects them.

Accel/Decel

The control can have any desired type of Accel/Decel ramping necessary for your machine.

Polar Co-ordinate Programming

An axis move need not be described by cartesian (rectangular) (XY) co-ordinates. The incremental radial distance (length) and an angle is all that is needed by the CNC, which automatically calculates where to move to. This often allows the operator to program direct from the blueprint. An arc can be specified by a radius, start angle, and end angle. You can mix polar and rectangular co-ordinate programming in an arc.

Feedhold

This allows the operator to stop machine motion at any time. "Cycle Start" will resume feed.

Block Delete Switch

The CNC will skip all lines of program that have the Block Delete code in the beginning of the line.

Resume Run

This powerful feature allows operator to interrupt his running program for tool inspection and resume where he left off. He would feedhold during a running program, then Jog or Handwheel the machine to a position for viewing the part. Upon pressing Resume and Cycle Start, the CNC will resume the original program from where Feedhold was initiated.

Optional Stop Switch

The CNC will insert a program stop if it encounters the Optional Stop code in the parts program.

E-Prom Memory

The executive control programs are burned into E-Prom chips, eliminating the need to load an executive tape. Just turn the power on and the CNC is ready to go.

Machine Parameter Selection

Specific machine parameters can be entered into a special machine system parameter memory at the time of setup to customize the CNC to a machine. Some of the parameters would select: Grid direction, rapid traverse rates, software overtravel limits, ballscrew and backlash compensation, RS-232 baud rate, unidirectional approach dimension, feedrate override selection, spindle interface, and paper interface selection. These will be retained by battery back-up or executive tape.

Spindle RPM Override and S-Word Analog*

The CNC can output a DC analog proportional to an S code. This code is used to control spindle RPM and can be made to work with nearly any spindle motor drive package. A spindle RPM override control allows the operator to modify the programmed spindle RPM.

Battery Back-Up*

This feature will prevent program loss or power loss. The CNC battery back-up will hold your program in memory for several years without charging.

CRT

The CRT is formatted to provide substantial programming assistance to the operator. This includes English description of M and G codes along with a comprehensive "Help" menu driven programming format. CRT position information can be magnified 2X or 4X for easier viewing. The CRT presents pertinent information like current position, next position, floating zero location, memory left, current block, tool offsets, diagnostics, and much more.

RS-232/TTY*

A bi-directional port is available to hook up to a TTY 20ma current loop or RS-232 port with adjustable baud rate. All popular personal computers or terminals can be interfaced to this port.

Magnetic Tape Storage*

An industrial cassette unit is built into many controls. A single cassette tape will store the equivalent of 500 feet of paper tape. Magnetic tape storage is low in cost, quick to access, and easy to edit. Magnetic tape can be used in conjunction with the RS-232 port and paper reader.

Paper Tape Reader*

The CNC can be equipped with paper tape reader of your choice. The CNC will take either EIA or ASCII format.

Machine "M" and "G" Codes

The control retains the ability to program in industry standard EIA/ASCII "M" and "G" codes or "Help" menu format. We believe that our CNC should have the best of both worlds. If the operator chooses to program the CNC using the "Menu" driven "Help" format he need not bother with any "M" or "G" codes. But if he decides to program in traditional N.C. language he is free to do so.

Ballscrew Backlash Compensation

When a code number is entered on the keypad, access is provided to a special system parameter memory. This allows the system to accept information to compensate for the backlash in each axis. This will correct for lost machine motion on reversal and is active in linear and circle interpolation moves.

BallscREW Error Compensation

By accessing the system parameter memory, through the keyboard, the operator can enter a ballscREW error compensation for all axes every .8192" apart. Nearly every style machine can position with extreme accuracy. Linear scales are seldom required for high accuracy positioning.

Unidirectional Approach

If selected, this feature results in all rapid axis moves to approach the final position from the same direction and at the same velocity to optimize positioning accuracy.

Parametric Programming (Creating your own macros)

This mode of programming allows the programmer to write a parts program using "Basic" like statements consisting of SIN, TAN, divide, multiply, IF, Then, GOTO, etc. to describe complex shapes. The CNC also has a library of different shapes with variable parameters. This gives the operator the "Fill In The Blank" concept and substantially reduces programming time and potential programming errors. Current shapes include sphere, triangle, triangle with draft, toroid, partial toroid, cylinder, bolt hole, rectangle, rectangle with radius corners, pocket clearing, etc.

Automatic Digitizing*

By adding the DNC Module and your off-line computer a probe can be installed in the spindle to automatically digitize models. After a model is digitized it can be played back to the CNC.

Fast-Cam*

The IIC Fast-Cam, Cad-Cam software system provides off-line computer aided programming with full graphics. The Fast-Cam is a 2½ axes Cad-Cam system developed especially for the IIC CNC package. Fast-Cam is easy to use and offers two way conversation to or from the CNC, island clearing, splining, and allows for modification of scaling, cutter compensating, and rotation of digitized programs.

Four Axes*

The Centurion IV can operate a true fourth axis servo which will interpolate with the primary three axes for rotary table control.

Automatic Cavity Clear

Saves an immense amount of programming time allowing the operator to clear out most irregular shaped XY cavity. It can also clear away from an island. The operator simply inputs the increment and number of passes.

Automatic Mill Cycle

A special canned cycle for incrementing the tool down to a final depth without the need of developing a subroutine. Simply instruct the CNC on first pass depth, increment, and final depth. This can be used in conjunction with Automatic Cavity Clear. Together they make a powerful combination for mold and tool and die work.

DNC Autolink*

Want to slave the CNC to your off-line computer? By the addition of the DNC "Autolink" module information can be received on-the-fly from the off-line computer. This results in your CNC memory size expanded to your off-line memory size.

Auxiliary M-Codes

Special function m-codes can be added to the CNC logic for implementation of single shot indexers or other on/off external accessories.

Automatic Tangent Radius

Automatically inserts a tangent radius between two lines, two arcs, or line arc combination. Only the radius needs to be specified. Saves programmer from doing complicated equations.

UPDATE TO THE CENTURION IV CNC CONTROL

JANUARY 5, 1989

1/5/89

Explanation of Software Enhancement Changes for the Centurion IV CNC Control

What follows is a listing of the additions and changes this software effects on the Centurion IV CNC control. Each feature has been explained so that the user can understand and make use of the new capabilities this software offers.

New Edit Features

By selecting "Display-Edit Program" in the "Help" mode, it is now possible to convert a main program into a subroutine. Follow the instructions on the screen to make the conversion.

When viewing a program or subroutine in the "Display-Edit" modes and the operator displays beyond the end of the program or subroutine it is now possible to go back into the program or subroutine by selecting the "+/-" key. Previous software required the operator to reselect the "Display-Edit" mode and call-up the desired event to be viewed or edited.

Parametric commands and other program code created without the use of the "Help" mode will now be displayed in the "Display-Edit" modes; although they cannot be edited in the "Help" mode. It is also possible to display program lines automatically created by the Centurion IV control, for the purpose of forcing program intersect points. These lines can be easily recognized by the " " character at the beginning of the line.

Entering or editing program zero is illegal. If this is attempted an error 10 will be displayed to indicate the illegal command.

RS-232 Changes

For those with computers hooked-up to the Centurion IV control via RS-232 there have been several changes. First, there is a change to the "Centurion IV Programming Manual" (reference the bottom of page 15-4). It is possible to "write" the Centurion IV system parameters into a computer file by performing the following steps:

1. Prepare the computer to receive a file via the RS-232. Make sure the baud rate of the computer is set the same as the Centurion IV control.
2. Select "Write Tape", 97, "ENT" at the Centurion IV control.

To send this file back to the Centurion IV control follow these steps:

1. Depress "Disp. Stat.", 9, "CE".
2. Depress "Disp. Stat.", 9086, "ENT", .0003, "ENT" (for a baud rate of 1200).
3. Now select "Read Tape", 97, "ENT".
4. Send the file created in the above procedure from the computer, via the RS-232, to the control. An "Error 03 Grid" will indicate that the parameters are loaded and the machine is ready to run.

If your machine has a paper tape reader make a note of the value in set-up parameter 86 ("Disp Stat", 9086, "ENT"). To load the set-up parameters from a paper tape follow these steps:

1. Select "Disp. Stat.", 9, "CE".
2. Depress "Disp. Stat.", 9086, "ENT", enter value of parameter 86, "ENT".
3. Insert the parameter paper tape in the tape reader and select "Read Tape", 96, "ENT". An "Error 03 Grid" will indicate that the parameters are loaded and the machine is ready to run.

When sending information to the Centurion IV control via RS-232 the control will now ignore all information prior to the "%" character. Previously, any characters prior to the "%" character would cause an "Error 08", or lock-up the control.

Although the Centurion IV control will not receive XON/XOFF characters, it will now send XON/XOFF characters. This means that programs longer than the control's memory can store, can be divided into several sections and sent to the control via RS-232 in the following way:

```
%N001 (E) (PS) G000 G017 X01.0000 Y01.0000 Z01.0000
.
.      (rest of program section 1)
.
N### (E) (MS) G55      (G55 calls for next section of program)
%***** ("*" characters used for delay
%N001 (E) (PS) G000 X02.0000      between "%" characters)
.
.      (rest of program section 2)
.
N### (E) (MS) G55
%*****
%N001 (E) (PS) G017 X00.0000 Y02.0000
.
.      (rest of program)
.
%      (end of program and transmission)
```

When a program is divided in this way and sent to the control it will automatically load the next section of the program as each section is completed.

General Control Feature Changes and Additions

The tool table has now been expanded to 40 tools from the 24 previously available. Upto 40 different tool offsets can be stored at the same time.

A M7## has been added to make it possible to automatically read a sector of tape and run the program stored there. For example, if a program is too large to fit into memory it can be divided into two or more sections. The first section could end with a M701, the second section could end with a M702, etc. When the first section of the program reaches the M701 command it reads sector one of the tape inserted in the tape reader and runs that section of the program until the M702 is encountered, which then causes sector 2 of the tape to be read and executed, and so on.

The pocket clear feature at the end of a mill cycle has been changed so that it clears the first depth entirely before going to the second depth and so on. It is also possible to enter a negative cut width so that an inside pocket can be cleared from the inside out, and an outside pocket cleared from the outside in. By using a tool radius offset larger than the actual tool size, this feature can be used to automatically accomplish a rough and finish cut of a profile. To do this, pocket clear would be selected on the tool retract screen at the end of the mill cycle, and then the number of passes would be set at 2 and the cut width at a negative value equal to the difference between the actual tool radius and tool offset radius.

Frame autoroutines, and the bolthole routine, use to cause the Z-axis to rapid to the R-plane before making a rapid move to the pierce point or first hole, and then feeding to the first Z depth. This sequence caused problems, and in many cases made these routines unusable if there was a clamp or other obstruction between the center of the routine and the pierce point or first hole. To correct this problem these routines now begin with a rapid move to the pierce point or first hole, then a Z rapid move to the R-plane, and finally a Z feed to first depth. At the end of the routine this sequence is reversed so that the Z-axis ends at the same location it was at when the routine was began. In this way obstructions can more readily be cleared without seriously affecting the run time of the routine.

The canned cycles have also been changed so that Z-axis moves can be programmed within the canned cycles without causing the cycle to be performed.

This makes it possible to clear a clamp within the drill cycle, for example, by simply making a positive Z-axis position move, to get the tool above the clamp, and then making a XY move to the next hole position; where the Z will rapid to R-plane before drilling the hole.

The wait channel used by the M32, M11-18, and M21-28 codes will now display a flashing "WAIT" when in use.

A M901 S#### has been added to the control's codes to make it possible to program a grid command for any axis or combination of axes. To determine the value of the S-code (####), for any grid command desired, refer to set-up parameter 066, bytes 2-5.

Additions and Changes to Parametric Programming

A new parametric print statement has been added, which allows the programmer to display the value of any parameter at any point in a parametric program. This can be a valuable debug tool for a programmer. To use it, one of the following statements is entered into the program:

```
(,? P##,) M0    or  
(,? P##,) G4 FXXX.X
```

The "?" is the print statement, the M0 and G4 commands are used to stop the program long enough to allow the programmer to read the parameter value when it is displayed on the screen.

Polar commands have been available for parametric programs but have not been shown in the manual. The format for a linear parametric polar move is as follows:

```
R00.0001 A000.002
```

Where the value of P1 is the magnitude of the line, and the value of P2 is the angle or direction of the line. The format for a parametric polar arc is as follows:

```
G2 R00.0003 A000.004 A000.005
```

where the value of P3 is the radius of the arc, the value of P4 is the start angle of the arc, and the value of P4 is the end angle of the arc.

In addition an arctangent function has been added to the parametric functions. For example:

```
P## = ATAN[P##]
```

Set-Up Parameter Changes

A Parameter has been added which allows all rapid moves to remain at 100% of the feedrate override setting regardless of what override has been selected. This allows the operator to slow cutting feeds without effecting the time required to make rapid moves.

This feature is not active in the Dry Run mode. To select this feature see set-up parameter 19 byte 2 (ref. pg. 6).

A maximum feedrate parameter has been added for each axis to limit its rapid or max. feedrate regardless of what feedrate is commanded. This allows different axes to have different rapid feeds for special applications. Refer to set-up parameters 37 thru 40 (ref. pg. 6).

Spindle commands can now be activated during the Dry Run mode. This allows the operator to see if the spindle is being turned on or off at the proper times in the program while checking the program in Dry Run. See set-up parameter 502 byte 1 (ref. pg. 11).

The Z-axis position display for "Current Position" and "Next Position" can now be referenced to Z zero for any tool length offset selected. For example: if the tool length offset is set for zero at the surface of the part being cut, then the Z-axis position will be referenced to the surface of the part rather than grid zero, providing set-up parameter 502 byte 1 has been entered properly (ref. pg. 11).

It is now possible to program a dwell before or after the spindle reverses direction or both during a tapping canned cycle. This makes it easier to program tapping cycles using floating tap heads. See parameters 90 byte 2 and 502 byte 1 (ref. pgs. 8 and 11).

Parameters have been added so that any axis or combination of axes can be gridded to the nearest encoder marker rather than the grid switch and then a marker. This can be useful in locating suspected faulty encoders, or in applications where it is difficult or impractical to install a grid switch. See set-up parameter 502 bytes 2 and 3 (ref. pg. 11)

Partial List and Explanation of System Parameters
for the
Centurion IV Control

For the following parameters with more than one function per byte, any combination of those functions can be set or selected by adding-up the number equivalent for the functions desired and entering the total into the parameter number at the location indicated by the "#" symbol.

Parameter 18 ####

Rapid feedrate parameter

0000 = default rapid feed of 400 IPM

2500 = 200 IPM

Parameter 19

Byte 0 000#

1 = reverse A-axis jog direction

Byte 1 00#0

1 = reverse X-axis jog direction

2 = reverse Y-axis jog direction

4 = reverse Z-axis jog direction

Byte 2 0#00

1 = feedrate at 100% for all rapid moves regardless of feedrate override switch position (does not affect Dry Run)

Parameter 20

Bytes 3210 ####

5000 = 100 IPM jog feedrate at 100% override setting

2500 = 50 IPM jog feedrate at 100% override setting

1000 = 20 IPM jog feedrate at 100% override setting

Parameters 037 - 040 are the maximum feedrate parameters. They are entered in IPM from 0 to 999.9 IPM, with 0000 selecting the default value of 400.0 IPM. An axis will not run faster than the value in this parameter regardless of the feedrate commanded or the value in parameter 018 (rapid feedrate).

Parameter 37 #### X-axis maximum feedrate parameter

Parameter 38 #### Y-axis maximum feedrate parameter

Parameter 39 #### Z-axis maximum feedrate parameter

Parameter 40 #### A-axis maximum feedrate parameter

Parameter 66

Byte 0 00000#

- 1 = enable (G12) manual digitize mode, otherwise disabled
- 2 = control will not initialize system flags before executing a program or MDI command (use M900 to initialize flags)
- 4 = will not compensate all subroutines after tool changes

Byte 1 0000#0

- 1 = automatic tool changer installed
- 2 = display will show following error during moves
- 4 = DNC memory cartridge is installed instead of magnetic tape reader

Bytes 5,4,3,2 ####00 are used to define the grid sequence as follows:

Byte 2 000#00

- 1 = grid X-axis
- 2 = grid Y-axis
- 4 = grid Z-axis

Byte 3 00#000

- 1 = grid A-axis

Byte 4 0#0000

- 1 = negative grid direction for X-axis
- 2 = negative grid direction for Y-axis
- 4 = negative grid direction for Z-axis

Byte 5 #00000

- 1 = negative grid direction for A-axis

Parameter 86

Byte 0 00.000#

- 0 = RS-232 set at 110 baud rate
- 1 = RS-232 set at 300 baud rate
- 2 = RS-232 set at 600 baud rate
- 3 = RS-232 set at 1200 baud rate

Bytes 1+2 00.0##0

- 40 = Autolink (DNC)
- 00 = EECO paper tape reader 150 cps
- 35 = Remex paper tape reader 300 cps
- 22 = Decitek paper tape reader 150 cps
- 23 = Decitek paper tape reader 300 cps
- 06 = EECO paper tape spooler 150 cps
- 07 = EECO paper tape spooler 300 cps

Parameter 86 (cont)

Byte 3 00.#000

- 0 = maintain the following outputs when commanded
- 1 = pulse coolant output
- 2 = pulse M11 output
- 4 = pulse M12 output

Byte 4 0#.0000

- 0 = maintain the following outputs when commanded
- 1 = pulse M13 output
- 2 = pulse M14 output
- 4 = pulse M15 output

Byte 5 #0.0000

- 0 = maintain the following outputs when commanded
- 1 = pulse M16 output
- 2 = pulse M17 output
- 4 = pulse M18 output

Parameter 88

Bytes 2,1,0 000###

- ### = number of steps divided by the maximum spindle RPM in low range - where steps equals 128 when spindle analog range from +5 to -5 volts or steps equals 256 when spindle analog range from 0 to 10 volts

Bytes 5,4,3 ###000

- ### = number of steps divided by the maximum spindle RPM in high range - where steps equals 128 when spindle analog is bidirectional or steps equals 256 when spindle analog range from 0 to 10 volts

Parameter 89

Bytes 2,1,0 000###

- ### = shift point RPM divided by 10

Bytes 5,4,3 ###000

- ### = maximum spindle RPM divided by 10

Parameter 90

Byte 0 00000#

- 1 = check 0 speed
- 2 = output creep speed
- 4 = check low range

Byte 1 0000#0

- 1 = check high range
- 2 = check spindle up-to-speed
- 4 = specify bipolar spindle D/A (-5 to +5 volt range)

Parameter 90 (cont)

Byte 2 000#00

- 1 = reverse M3/M4 in low range
- 2 = level M3/M4
- 4 = dwell for number of seconds loaded into program parameter 23 after spindle reverses direction in tapping canned cycle

Bytes 4,3 0##000

- ## = creep speed in DAC (digital to analog) units

Byte 5 #00000

- 1 = activate spindle speed override potentiometer
- 2 = make M-functions go into handshake mode if a programmable controller is used with the CNC
- 4 = system will not cutter compensate subroutines before compensating the main program (must use M6XX or M600 to compensate subroutines)

Parameters 486-491 are the power-up flags. These flags will only be active if parameter 490 byte 3 (#000) bit 2 is set (IE 490 = 4XXX, 5XXX, 6XXX, or 7XXX). Note that these parameters work in pairs, 486 and 487, 488 and 489, 490 and 491, meaning that a desired setting may be made in one, or the other as needed.

Parameter 486 or 487

Byte 1 00#0

- 1 = axis interchange
- 2 = mirror S2-axis
- 4 = mirror S1-axis

Byte 2 0#00

- 0 = XY plane
- 1 = XZ plane
- 2 = YZ plane
- 3 = XC plane
- 4 = YC plane
- 5 = ZC plane

Byte 3 #000

- 1 = mirror X-axis
- 2 = mirror Y-axis
- 4 = mirror Z-axis
- 8 = mirror A-axis

Parameter 488 or 489

Byte 0 000#

- 2 = rapid feed mode

Parameter 488 or 489 (cont)

Byte 1 00#0

1 = grid required

Parameter 488 or 489

Byte 2 0#00

1 = right tool radius compensation (G42)

4 = clockwise circular interpolation (G2)

Byte 3 #000

4 = continuous path mode (G30)

8 = metric mode (G71)

Parameter 490 or 491

Byte 0 000#

1 = enable S2-axis

2 = enable S1-axis

4 = enable C-axis

8 = enable Z-axis

Byte 2 0#00

1 = sense hi-low spindle range

4 = accel/decel on

8 = accel on

Byte 3 #000

1 = display C-axis

4 = enable power-up flags

Parameter 492

Bytes 3,2,1,0 ####

0000 = fastest accel/decel slope possible

0118

1220

2324

0030

3240

4250

3160

6270

5190 = slowest accel/decel slope possible

Parameters 493 to 498 ####

These parameters control grid speed to and away from the grid switch (EG. 0000 = default value (60 IPM)

0412 = 60 IPM
0618 = 90 IPM
0206 = 30 IPM)

493 = X-axis grid speed
494 = Y-axis grid speed
495 = Z-axis grid speed
496 = A-axis grid speed
497 = S1-axis grid speed
498 = S2-axis grid speed

Parameter 499 ####

This parameter controls the grid speed towards the marker for all axes. The larger the value of this parameter the faster the speed to the marker.

0000 = default speed
0060 = 8.5 IPM

Parameter 500 ####

Dry run feedrate in IPM (0 - 999.9 IPM)

0000 = default feed of 75 IPM
1000 = 100 IPM
0500 = 050 IPM

Parameter 501 ####

Maximum programmable feedrate (0 - 999.9 IPM)

0000 = default max. feed of 100 IPM
1000 = 100 IPM

Parameter 502

Byte 0 000#

- 1 = spindle stays on at end of program
- 2 = grid Z-axis first
- 4 = spindle on during Dry Run

Byte 1 00#0

- 1 = accel/decel ramps turned off
- 2 = display Z-axis position relative to tool tip at Z zero
- 4 = dwell for number of seconds loaded into program parameter 23 before spindle reverses direction in tapping canned cycle

Byte 2 0#00

- 1 = grid X-axis to nearest encoder marker
- 2 = grid Y-axis to nearest encoder marker
- 4 = grid Z-axis to nearest encoder marker

Byte 3 #000

- 1 = grid A-axis to nearest encoder marker

M-FUNCTIONS

1.2

M0	Program Stop
M1	Optional Stop
M2	End of Program
M3	Spindle on CW
M4	Spindle on CCW
M5	Spindle Stop
M6	Tool Change (depress TOOL CHANGE RESET button to clear this code once activated)
M7	Mist coolant on
M8	Flood Coolant on
M9	Coolant Off
M11-M18	Turn Channels 1-8 on
M21-M28	Turn Channels 1-8 off
M30	Rewind Paper Tape
M32	Wait Channel
M600	Forces all subroutines to be re-cutter compensated, used after a tool change or G20, G21 command.
M6XX	Forces subroutines specified by XX to be re-cutter compensated, used after a tool change or G20, G21 command.
M999	Same as "/" block delete code, if block delete switch is pushed block containing the M999 will not be executed.
M998	Code that defines the start of a milling cycle, used by help.
M997	Code that defines the end of a milling cycle, used by help.
M996	Code that defines a non-move block, X and Y dimensions are to be used by cutter compensation only.
M995	Preceeds a feedrate, or spindle command in parametric mode, the F or S command will then use parameters for their values.
M994	Retract Z to tool change position.

G-CODES

G0 RAPID - Initiates rapid feedrate unidirectional approach
 G1 FEED MODE .1 to 100 ipm (set by "F" code) cancels rapid
 G2 CW Circle Definition (modal)
 G3 CCW Circle Definition (modal)
 G4 FXXXX.X causes dwell for XXXX.X seconds
 G10 Display Z Axis (modal)
 G11 Display C Axis (modal)
 G12 FXXXX Digitize starting at block XXXX
 G13 XC Plane Circle (modal)
 G14 YC Plane Circle (modal)
 G15 ZC Plane Circle (modal)
 G17 XY Plane Circle (modal)
 G18 XZ Plane Circle (modal)
 G19 ZY Plane Circle (modal)
 G20 Add/Subtract to Tool Offsets
 G21 Enter Tool Offset Table
 G26 Autoroutine: Circular or Rectangular Finish Cut
 G27 Autoroutine: Rectangular Pocket Clearing or Symmetric Hole Drilling Pattern
 G28 Spiral Pocket Clearing (enter increment with G305 PXX.XXXX)
 G29 G29 followed by PXX.XXXX-sets R-plane at XX.XXXX inches
 G30 Continuous Path Mode (modal)
 G31 Point to Point - Axes, make complete stop after every move (modal)
 G32 Cancel G92 (Floating Zero)
 G40 Turns Tool Radius Compensation Off
 G41 Turn on Left Tool Radius Compensation
 G42 Turn on Right Tool Radius Compensation
 G50 Rewind Paper Tape
 G51 Read Paper Tape
 G52 Wait Channel 18
 G53 Memory to RS-232 EIA
 G54 Memory to RS-232 ASCII
 G55 RS-232 EIA/ASCII to Memory
 G58 Output Leader to RS-232
 G60 Mirror Image Off (all axes)
 G61 X Mirror Image On (modal)
 G62 Y Mirror Image On (modal)
 G63 Z Mirror Image On (modal)
 G64 C Mirror Image On (modal)
 G65 X-Y Axis Interchange (modal)
 G70 English Mode (modal)
 G71 Metric (feedrate remains in mm per minute)(modal)
 G79 Re-enable Canned Cycle
 G80 Turns Off Canned Cycle G81-G87
 G81 Drill Canned Cycle
 G82 Drill with Dwell (G323, P2 = 2 sec. dwell)
 G83 Peck & Drill (G306 and G307 + P dimensions enter peck increments)
 G84 Tap Canned Cycle
 G85 Bore Canned Cycle
 G86 Mill Canned Cycle
 G87 Peck & Drill (does not return to R-plane)
 G89 Bore with Dwell Canned Cycle
 G90 Absolute (modal)
 G91 Incremental (modal)
 G92 Resets Floating Zero at Entered X,Y,Z,C Values
 G98 Return X, Y, Z to Zero
 G99 Return to R-plane

G-CODES (Cont.)

G100	Deactivates Tool Offsets
G1nn	Activates Tool Offsets X, Y, Z for Tool nn
G3xx	Enter Parameters
G4xx	Adds and Subtracts to Parameters
G500	Switch to parametric mode (see Section XIV on parametrics)
G501	Exit parametrics mode (see Section XIV on parametrics)
G7xx	Define Subroutine
G8xx	Call Subroutine
G8xx	LYY Loop on Subroutine XX, YY Times
G9xx	Bolt hole

ERROR CODES

1.4

<u>ERROR#</u>	<u>Explanation</u>
00	Attempt was made to run program 0 or 9 (9001, 9002, etc.). Program 9 is for subroutines only. Program "0" does not exist.
01	Attempt to run or DISPLAY/EDIT a block or program which does not exist.
02	The Auto Lube sensor indicates a low level of lubricant.
03	Attempt made to move an axis before the machine was gridded. (JOG & H.WHEEL are exempt)
04	The Position encoders detected an excess following error; or, the drives are off during a move command (press Reset); or Emergency stop was pushed during a move; or Feed Hold pushed during Grid.
05	A block was input which is longer than allowed.
06	The spindle sensor indicates it is out of gear. (Parameters SP 88 Thru SP 90 requested this input be checked).
07	The system was corrupted (by power down of Battery Back-up failure). The machine set-up table (parameters "SP") must be loaded.
08	A magnetic tape was read in error: usually means it was written on a system with less/more memory. (To prevent error 18 after this, press 'CLEAR PROGRAM' and the block # +1 of the last subroutine block, or 9001 if no subroutines); else, ERROR 8 means the tape was corrupted.
09	Attempt to make a magnetic tape has failed; try again. If repeated failure, try another tape (memory is still OK).
10	The Edit Key is off when an attempt is made to modify problems in memory.
11	No tape in the tape reader when 'READ TAPE' or 'WRITE TAPE' is executed.
12	Attempt made to 'READ TAPE' when the safety tab is in the 'UNPROTECTED' position.
13	Attempt made to "WRITE TAPE" when the safety tab is in the "PROTECTED" position.
14	Tape sector too large. The maximum number of tape sectors depends on the size of memory.
15	Invalid command. A general error indicating that accepted protocol was violated. Refer to the manual for description of the particular case in which the error occurred.
16	Attempt was made to read or write to a magnetic tape which was not formatted as described in the diagnostics section of this manual titled Erase Magnetic Tape.
17	The tool number set by G1XX or referenced by G20, G21 or Display Status Page 6 must be less than 25 to prevent this error.
18	Memory Full. This will occur when the memory is too full to be expanded or when some "Garbage" has been found at the end of memory. To rectify the latter, use "Clear Program 9nnn", where nnn is the last block of the subroutines +1, or 9001 if there are no subroutines. If memory really is full, this error appears when cutter comp. wants to insert a block because of "(CGEN)", "(RNDC)", or when two arcs don't intersect.

ERROR #	Explanation
19	Grid was attempted with No axis enabled. Machine configuration Parameter SP 066 cannot be zero.
20	Spindle overspeed error. Check system parameters 88, 89, 90.
21	A program contains an erroneous application of the circle generate function. Probably, there are not 3 blocks of linear points following the "(CGEN)" block.
22	A program has requested an impossible line-to-circle or circle-to-line intersection. This may mean the cutter radius is too large, or that "trig-help" is needed (it is enabled by a G41 or G42, and disabled by G40).
23	A program has requested an impossible circle to line intersection. See error 22.
24	The Drives have not shut off. CNC cannot digitize.
25	The block number for digitize (G12), or the number of seconds for delay (G4) is missing in the block.
26	Cutter comp. intersection error in an Autoroutine. This should never happen. If it does, Bad ROM probable.
27	Cannot Resume. This error occurs when the system has no location at which to resume the program.
28	A program contains on erroneous application of the "Round corner function". Probably there are not 2 lines or circular definitions proceeding the (RNDC) block and another linear or circular definition following the (RNDC) block.
32	The handshake mode has timed-out.
40	Software limit overtravel. An arc block will end with the cutter outside the limit on one or more of the axes.
41	Software limit overtravel. The X axis command is beyond the machine limit.
42	Software limit overtravel. The Y axis command is beyond the machine limit.
43	Software limit overtravel. Both X and Y axes commands are beyond limits.
44	Software limit overtravel. Z axis command is beyond machine limit.
45	Software limit overtravel. X and Z.
46	Software limit overtravel. Y nad Z.
47	Software limit overtravel. X,Y, and Z.
48	Software limit overtravel. A axis is beyond the machine limit.
49	Software limit overtravel. Both X and A.
4:	Software limit overtravel. Y and A.
4<	Software limit overtravel. Z and A.
4=	Software limit overtravel. X, Z, and A.
4>	Software limit overtravel. Y, Z, and A.
4?	Software limit overtravel. X, Y, Z, and A.
50	End point error. A move block failed to arrive at the expected end point. Most often caused by incorrect circle center or radius, or by loss of the drives during the move.
58	The Helical axis (in a 3-axis circular move block) has been requested to move too short a distance for a very large arc radius.
Parametric Programming ERRORS; 60 THRU 74	
60	The label indicated in a parametric GO TO or IF statement cannot be found elsewhere in the program.
61	The label indicated in a parametric GO TO or IF statement was found, but only in another program. Jumps must be within the same program.
62	A label is missing or misplaced in this block. eg., ",L1, GO TO,)" instead of the correct form: "(,GO TO L1,)"
63	An unrecognized operation was found, eg. "(,P1=P2% P3,)" or parameter number was missing after a "P" eg. "(,P1=P2/P,)"

<u>ERROR</u>	<u>Explanation</u>
64	Brackets are unbalanced (the number of left brackets "[" is not equal to the number of right brackets "]"0, eg: "(,P1=[P2* [P3 + P4[,)
65	Parametric line too long. Maximum length is 64 characters per block.
66	Syntax error. There are missing (or excessive) operands or operators. Eg: (,P1=P2 P3,) (,P1=P2-*P3,)
67	A label is missing the number. For example, "(L,)" instead of "(,L3,)"
68	Literal string is too long. A number in an equation, such as (,P3=3.141592654,) will give this error if it is more than 8 digits long.
70	Overflow. A calculation has produced a result which is greater than the maximum allowed. ($1.3 * 10^8$).
71	A division by 0.0 or the TAN(90) has been attempted.
72	A square root has been tried with a negative number.
73	A trigonometric function has been given an erroneous input value.
74	An unrecognized operation has been discovered.
77	In an autoroutine, the parameter 'RADIUS' is smaller than the cutter radius.
7>or7?	Autoroutine memory is corrupted.
80	Parameter 3(radius) is greater than parameter 1 or 2 in a rectangular pocket autoroutine: or Parameter 5 is zero in G27 or G28.
81	Angle of first hole in bolthole pattern is negative. This should always be a position angle.
85	No X, Y, Z, or R dimension after a G20 or G21 command.
87	Autoroutines error: incorrect parameter values.
88	Program cannot be displayed in Help mode. It was probably entered or edited in the standard EIA program mode of operation.
94	Duplicate subroutine numbers.
95	Programmed subroutine does not exist.
96	Subroutine start $\neq$ subroutine end.
97	Subroutine never terminated.
98	Exceeded subroutine nest limit.

1.5

MACHINE COMMANDS

<u>FORMAT</u>	<u>DESCRIPTION</u>	<u>MAXIMUM VALUE</u>
M3	Miscellaneous Code	M999
F3.1	Feedrate (IPM)	F999.9
G3	Preparatory Code	G999
R2.4	Radius	R99.9999
S4	Spindle Speed (RPM)	S9999
X <u>2</u> .4	X-Axis	X <u>99</u> .9999
Y <u>2</u> .4	Y-Axis	Y <u>99</u> .9999
Z <u>2</u> .4	Z-Axis	Z <u>99</u> .9999
C <u>3</u> .3	C-Axis*	C <u>99</u> .9999
Z/K <u>2</u> .4	Parameter	P <u>99</u> .9999
G323 A/K2.1	Dwell Parameter	Z/K99.99

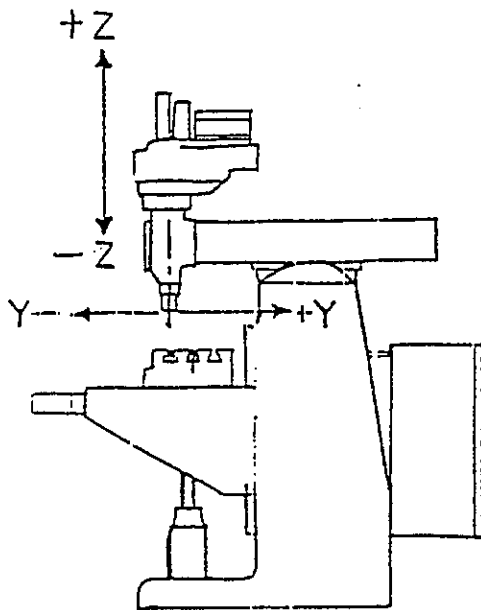
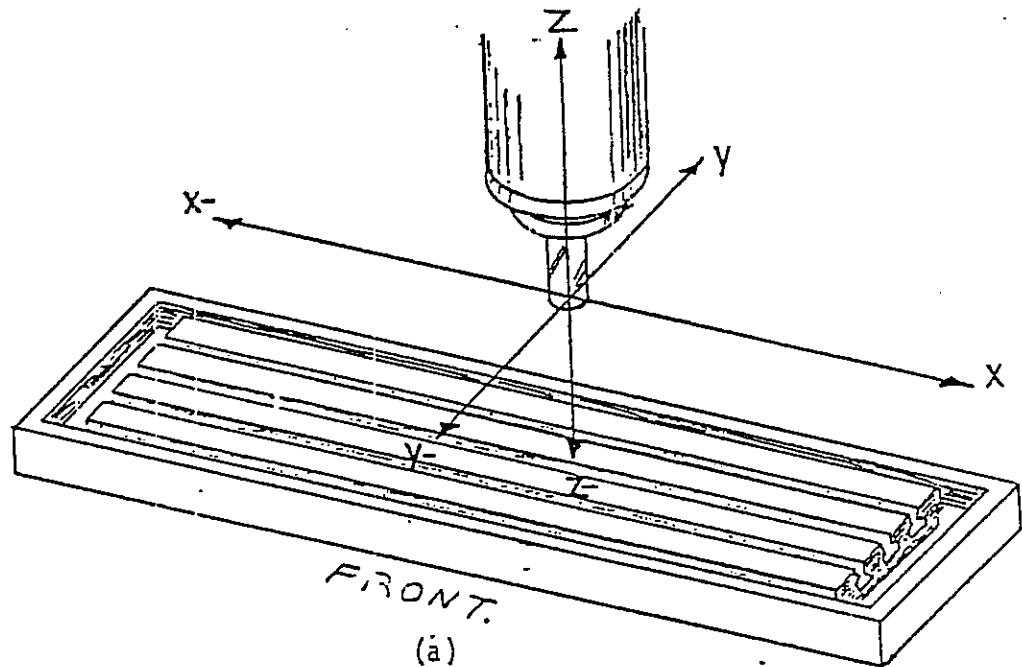
METRIC FORMAT

M3	Miscellaneous Code	M9999
F4	Feedrate (MMPM)	F9999
G3	Preparatory Code	G999
R3.3	Radius	R999.999
S4	Spindle Speed (RPM)	S9999
X <u>3</u> .3	X Axis	X <u>999</u> .999
Y <u>3</u> .3	Y-Axis	Y <u>999</u> .999
Z <u>3</u> .3	Z-Axis	Z <u>999</u> .999
C <u>3</u> .3	C-Axis	C <u>999</u> .999
Z/K <u>3</u> .3	Parameter	P <u>999</u> .999

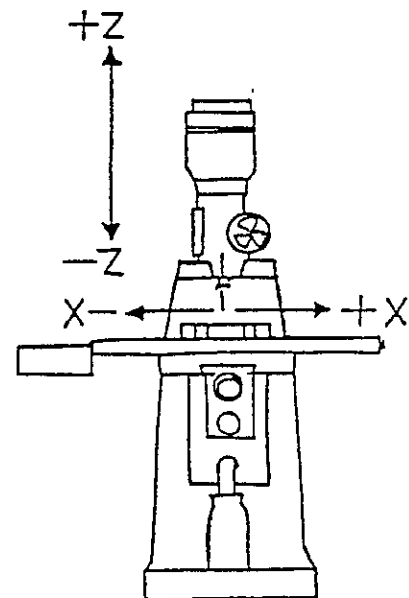
*The C-axis could also be an A or B rotary axis.

1.6 AXIS DEFINITIONS

All directions are referenced with respect to the tool. The following illustrates the X, Y, Z directions.



(b)



(c)

SECTION II

OPERATOR'S CONTROL CONSOLE

2.1 PANEL LAYOUT

The keyboard for the Centurion control has been specifically designed and organized to make this control the easiest and most efficient to learn, use and operate. There is no shift key, meaning each key has a specific purpose. This makes it possible to label each key as to its specific function for easy reference. In addition to this, the keys are organized in sections with related key functions being grouped together. The CLEAR ENTRY[CE] and ENTER keys are kept separate for easy access. The numeric, plus, minus, and decimal keys make up the numeric section of the keypad. The machine "instruction" keys and MDI key are located along the top and left edges of the 24 key section to the left of the numeric section. The "instruction" keys are arranged for easy access, since these keys are more frequently used. They are located along the edge so that an LED indicator could show operation of these keys. The remaining 15 keys are primarily mode of operation keys and are divided into two more sections. The six keys on the left (HANDWHEEL ENABLE, DRY RUN, RAPID JOG, GRID, JOG AND RUN) and the MDI key determine the mode of axis move to be made (Handwheel Enable, Dry Run, Jog, Grid or MDI). The other 9 keys apply to program entry and the edit capabilities of the control. This arrangement makes the keys easy to locate and minimizes confusion when entering information.

2.2 CONTROL CONSOLE SWITCHES

2.2.1 PUSHBUTTONS AND SELECTOR SWITCHES

A. FEEDRATE OVERRIDE SELECTOR SWITCH

Offers to the operator the ability to select any of nine different percentages of the programmed feedrate. The nine increments available are: STOP-10, 30, 70, 90, 100, 110, 130, 150 percent. In jog mode feedrate override selects the above percentages of the last feedrate for jog feedrate. In electronic handwheel mode, angular rotation of handwheel is variable in the nine selectable increments, for a resolution of .0001" linear motion of selected axis. This switch also has incremental jog settings of .0001", .001", .01", and .1".

B. AUTO/BLOCK SELECTOR SWITCH

An automatic program stop is inserted after each block of the stored program when selected to "BLOCK". When selected to "AUTO", continuous operation results with program stops occurring as written.

- C. EDIT KEY ON/OFF SWITCH
Edit key "ON" is normal operation. Edit key "OFF" disables keyboard program changes, or new program to be read into memory by magnetic or punched tape.
- D. SPRAY COOLANT/OFF/AUTO/FLOOD SELECTOR SWITCH
Provides for manual selection of either spraymist, flood coolant or coolants off. In "AUTO" position these same selections can be controlled by programmed "M" codes.
- E. SPINDLE STOP PUSHBUTTON
Available to give operator control for manual spindle stop. A programmed "M" code can also be used.
- F. SPINDLE CCW/CW SELECTOR
A spring return to center position switch to manually start the spindle in either the CW or CCW direction. Programmed "M" code can also be used to start the spindle CW or CCW.
- G. RESET
Enables the axes drives, the spindle start and coolant selector switches.
- H. EMERGENCY STOP PUSHBUTTON
Disables the servo drives, the spindle motor and the coolants.
- I. TOOL CHANGE RESET
When tool change "M06" is programmed the spindle is shut off and locked out, allowing for a safe tool change. This button must be pushed and held in for 1 second and the spindle started to continue the programmed cycle.

2.2.2 KEYPAD KEYS

- A. ELECTRONIC HANDWHEEL (EN HDW)
Activates the electronic handwheel mode. The X, Y, Z, C keys on keypad must be selected for the axis which is to move. The control displays position from the last floating zero and operates as a DRO unit.
- B. ELECTRONIC HANDWHEEL (optional) (not on keypad)
When turned moves the axis selected directly proportional to speed and direction of handwheel being turned. The angular rotation of handwheel for .0001" linear motion resolution is variable, by the feedrate override selector switch. Continuous axes position readout is present.

A program can be made by moving to position with the handwheel. (See programming by handwheel.) Electronic handwheel is also useful for setting of floating zero positions or tool offset positions.

C. COMMAND KEYS

1. MDI - "Manual Data Input" made for single block operation. An entire line of information can be inserted into memory including linear and circular moves and "M" or "G" functions for a one time move.
2. M - Miscellaneous function programming of machine commands for control of coolant, indexer, spindle, etc.
3. R - Radius entry key for radii of circular interpolation mode of operation with radii up to 99.9999 inches. Also for entering tool radii.
4. X/i - X axis programming commands, selecting jog or handwheel operation of axis.
5. Y/j - Y axis programming commands, selecting jog or handwheel operation of axis.
6. Z/k/P - Z axis programming commands, selecting jog or handwheel operation of axis. Also entering parameters (P).
7. C - Rotary axis programming commands, selecting jog or handwheel operation of C, A, B axis.
8. S - "S" code for spindle rpm command of variable speed drives.
9. F - Axis feedrate command from .1 ipm to 100 ipm (for .0001" resolution).
10. G - Codes for circle direction, tool compensation commands, mirror image commands, floating zero commands, canned cycle commands and other miscellaneous control commands.
11. GRID - When selected, all axes automatically go to their home reference position. This position is usually defined by a limit switch and encoder marker pulse.

12. JOG

PRESS [JOG]

The numeric keypad now becomes a "joystick" (see diagram).

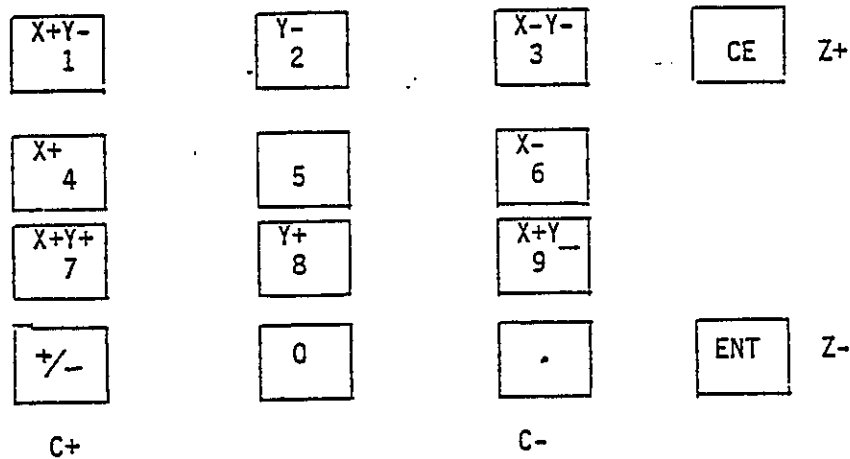


FIGURE 2.2-1 NUMERIC KEYPAD SHOWING JOG COMMAND KEYS

1. Key causes positive X and negative Y motion
2. Key causes negative Y motion
3. Key causes negative X and negative Y motion
4. Key causes positive X motion
6. Key causes negative X motion
7. Key causes positive X and negative Y motion
8. Key causes positive Y motion
9. Key causes negative X and positive Y motion
- (+/-) Key causes positive C motion
- (.) Key causes negative C motion
- CE Key causes positive Z motion/ENT key causes negative Z motion

When feedrate pot is between 10-150% the machine will move as long as a key is pushed down. Incremental jog becomes active when the pot is moved to the .1, .01, .001, .0001 positions and a key is pressed for direction and axis.

13. DISPLAY STATUS - When initiated and followed by a numeric "page" number will display present position from floating zero, floating zero from absolute zero, present position from absolute zero, programmed feedrate, active tool number, spindle rpm, I/O status, G codes status, parameter values, and much more. (Parameters are explained in Section II, 2.5)
14. CLEAR PROGRAM - When initiated and followed by block number erases program from memory.

15. DELETE - By initiating, deletes that information from a block which the operator has on display.
16. END OF BLOCK - Indexes the block number and stores the line information in the display when writing a program.
17. DISPLAY PROGRAM - When initiated and followed by program and line number displays program data as ENTER key is incremented. At any given time "delete" can be initiated to remove displayed information or new information can be entered for editing. The new information will be displayed also. Additional entries are just entered at the end of the line. To write a new program select the program and block number, enter in information, and end each block with "End of Block" key.
18. EDIT ADD - Initiated to add new blocks (lines) of information in a program. The entire program is automatically indexed when lines are inserted.
19. READ TAPE - When initiated and followed by a two digit sector number (01 through 10) control will read sector selected from the magnetic tape cassette. Number of tape sectors is determined by the amount of memory in the control, ie: the more memory the fewer sectors on a tape.
20. WRITE TAPE - When initiated and followed by a two digit sector number (01 through 10) control will write the entire contents of the memory on the sector of magnetic tape, as called for by number. Sectors and programs within sectors can be edited at the machine by reading them into memory and using editing procedures. The edited sectors and programs are written over the old sectors within the magnetic tape cassette, so old programs are destroyed. All cassettes have a safe switch on them so sectors and programs cannot accidentally be written over.
21. CE-CLEAR ENTRY - Clears any data shown in readout without clearing previously entered block data. Note once data has been accepted by entering into memory it can only be changed by editing procedures.
22. ENTER - Transfers information from display to memory
23. HELP - Used in MDI and DISP PROG modes to display a directory of programming aids for use by the operator. This "HELP" includes aids for programming arcs, rectangular pocket clearing, rectangular or circular finish cuts, circular pocket clearing, bolt hole routines, and Z-axis canned cycle routines.

2.2.3 NUMERIC KEYPAD

1. +/-
Changes state from "+" to "-" or vice versa when programming or moving about a floating zero.

2. . Inserts the decimal point when initiated. If not initiated by the third numeric input, the decimal is automatically inserted after the second whole number in inch mode and the third whole number in metric.
3. 0-9
Numeric numbers for all commands.

2.3 DISPLAY STATUS FEATURES

The DISPLAY STATUS feature enables the operator to view a considerable amount of valuable information stored in the control. This information is available to the operator on nine display "pages". These "pages" are numbered 0 through 7 and 9. The information displayed on each "page" is listed below, by page number.

KEY SEQUENCE

DISP
STAT , 0

<u>CURRENT POSITION</u>	<u>NEXT POSITION</u>	<u>FLOATING ZERO</u>	<u>STATUS</u>
X-000.000	X	X 003.102	Left Cut
Y-003.000	Y	Y-010.588	CCW
Z 000.000	Z	Z 000.000	Inc/Metric

<u>BLOCK #</u>			
r-plane: 000.000			
Auto :			
Cycle :	G86	(inactive)	
Feed :	020.0	mmpm (Mode-Feed)	Dwell: 02.0 sec
Tool # :	00	(offsets inactive)	Axis Inter:
Tool R :	000.000		Plane : XY
Spindle:	0000rpm		Mirror: Y
Z-Depth:	000.000		

FIGURE 2-3.1 CURRENT DISPLAY STATUS

The display status shows the current status of the control at any given time. The top portion of the display shows position data.

CURRENT POSITION - Present location relative to set zero(Floating Zero).

NEXT POSITON - Next location called for by program or in MDI relative to set zero.

FLOATING ZERO - Program zero referenced to grid or machine home position.

STATUS -

Left Cut (or Right Cut) - Direction of cutter compensation relative to cut direction.

CCW (or CW) - Counterclockwise or clockwise circular interpolation.

Inc (or Abs) - Incremental or absolute mode.

Inch (or Metric) - Available modes.

The bottom portion of this display shows program data.

R-PLANE - R-plane is a reference plane established by the programmer above the work piece that is free of obstacles. The control will make automatic moves in RAPID above this plane. (See section on canned cycles.)

AUTO - This shows current autoroutine in progress. (See section on autoroutines.)

CYCLE - This will show the current canned cycle in operation, and whether it is active or not.

FEED - This shows current feedrate in either inches per minute (ipm) or millimeters per minute (mmpm) and indicates whether the FEED mode or RAPID mode is active.

TOOL # - This shows last tool # selected and whether fixture offsets are active or inactive.

TOOL RADIUS - This shows the current tool radius and whether cutter radius compensation is turned on or not.

SPINDLE - This shows the most current spindle S-word command executed by the control.

Z-DEPTH - This shows the current depth parameter value.

DWELL - This shows the countdown to zero of the dwell time as a dwell instruction is executed by the control.

AXIS INTERCHANGE - This shows if an axis interchange instruction is active (G65).

PLANE - This shows the current plane for circular interpolation.

MIRROR - This shows all the axes that are mirror imaged.

DEFINITION OF DISP STATUS PAGES

2.3.2 1. G-CODE LISTING

G0 RAPID	G4 DWELL	G14 YC PLANE	G19 YZ PLANE
G1 FEED	G10 Display Z AXIS	G15 ZC PLANE	G20 ADD/SUB T. OFF
G0 CW	G11 DISPLAY C AXIS	G13 XY PLANE	G21 LOAD T. OFFSET
G3 CCW	G13 XC PLANE	G18 XZ PLANE	G26 RECT. POCKET
G40 TOOL RAD. COMP. OFF	G61 X MIRROR ON	G27 FACING CYCLE	
G41 LEFT TOOL RAD. COMP.	G62 Y MIRROR ON	G28 SPIRAL POCKET	
G42 RIGHT TOOL RAD. COMP.	G63 Z MIRROR ON	G29 SET R PLANE	
G60 MIRROR IMAGE OFF	G64 C MIRROR ON	G65 AXIS INTER	
G70 ENGLISH MODE	G81 DRILL CYCLE	G85 BORE	
G71 METRIC MODE	G82 DRILL W/DWELL	G86 MILL	
G79 RE-ENABLE CAN CYCLE	G83 PECK DRILL	G87 PECK DRILL II	
G80 DISABLE Z CYCLE	G84 TAP	G89 BORE W/DWELL	
G90 ABSOLUTE	G98 RETURN XYZ TO ZERO	G3XX LOAD PAR. VALUES	
G91 INCREMENTAL	G99 RETURN Z TO R-PLANE	G4XX ± PAR. VALUES	
G92 PRESET XYZC	G100 CANCEL TOOL OFFSETS	G7XX DEFINE SUB.	
G9XX BOLT HOLE	G1XX ACTIVATE TOOL OFFSETS	G8XX CALL SUBROUTINE	

FIGURE 2.3.2

CRT DISPLAY OF G-CODE LISTING PAGE 1

2.3.3 2 G-CODE LISTING CONT.

G50	REWIND PAPER TAPE	G54	MEM TO RS-232 ASCII	G31	POINT/POINT
G51	READ PAPER TAPE	G55	RS-232 EIA/ASCII - MEM	G32	CLR F ZEROES
G52	WAIT CH. 18	G58	OUTPUT LEADER - RS-232		
G53	MEM TO RS-232 EIA	G30	CONTINUOUS PATH MODE		

FIGURE 2.3.3

CRT DISPLAY OF G-CODE LISTING PAGE 2

2.3.4 3 M-FUNCTION LISTING

M0	PROG. STOP	M4	SPINDLE CCW	M8	FLOOD
M1	OPTIONAL STOP	M5	SPINDLE OFF	M9	COOLANT OFF
M2	END OF PROG.	M6	TOOL CHANGE	M10-M99	OUTPUT TO BUSS
M3	SPINDLE CW	M7	MIST		

FIGURE 2.3.4 CRT DISPLAY OF M-FUNCTION LISTING

Page #4 is used for additional M-Function listings when necessary.

2.3.5 I/O CHANNELS - Page #5

DISP 5
STAT

This display is used by the operator or maintenance personnel for trouble shooting machine problems. It displays inputs and outputs of the system. This page is exited by depressing "Cycle start" which returns us to Display Status page 0.

MEMORY LEFT: 15203 LOCATIONS			CHK 42B5		
GRID	MARKER	'O' ERROR	SOFTWARE LIMITS		COMP #
X 1	1	0	00.5000	-29.5000	0100
Y 0	1	1	00.0000	-11.7500	0135
Z 1	1	1	00.0000	-05.0000	0149
A 1	1	1	00.0000	-00.0000	0150
B					
C					

PARALLEL INPUTS	00-07	0000	0000	0000	0000	0000	0000	0000	0000	0000
	08-0F	0000	0000	0000	0000	0000	0000	0000	0000	0000
	10-17	0000	0000	0000	0000	0000	0000	0000	0000	0000
	18-1F	0000	0000	0000	0000	0000	0000	0000	0000	0000
	20-27	0000	0000	0000	0000	0000	0000	0000	0000	0000
	28-30	0000	0000	0000	0000	0000	0000	0000	0000	0000
SERIAL INPUTS	00-1F	0000	0000	0000	0000	0000	0000	0000	0000	0000

FIGURE 2.3.5

I/O CHANNELS

The following is an explanation of the information found on this page.

MEMORY LEFT - Number of bytes or entries remaining in memory for parts programs.

CHK - This is a parity check on the RAM memory - if this number changes after a master clear or power down it means the memory has been changed and should be checked or reloaded.

PAGE #5 CONT.

GRID - Normally a "1" for all axes which switch to a "0" when the grid switch for that axis is made.

MARKER - Normally a "1" for all axes which switch to a "0" when the encoder marker is detected.

'0' ERROR - Not used.

SOFTWARE LIMITS - These are the software limits entered in machine parameters 006 to 017.

COMP # - this is the beginning lead screw error compensation machine parameter for each axis.

PARALLEL INPUTS

Channel 00 - Keyboard (upper characters)
Channel 01 - Keyboard (lower characters)
Channel 02 - Paper Tape (upper 4 bits)
Channel 03 - Paper Tape (lower 4 bits)
Channel 04 - Feedrate Override
Channel 05 - Handwheel (encoder)
Channel 06 - Joystick] Jog
Channel 07 - Joystick]
Channel 08 - Spindle Override
Channel 09]
.
.
.
.
OF]
- Digitize Inputs

SERIAL INPUTS

Channel 00 - File Protect
Channel 01 - Read Data
Channel 02 - End of Tape (EOT)
Channel 03 - Carry
Channel 04 - P Key (on black box)
Channel 05 - Initiate (on black box)
Channel 06 - Tape Ready
Channel 07 - Data Available
Channel 08 - Sprocket
Channel 09 - Edit Key
Channel 0A - Auto/Block
Channel 0B - Key Pushed
Channel 0C - CRT Timer Pulse
Channel 0D - S Zero Marker
Channel 0E - Reset Ground
Channel 0F - Optional Stop
Channel 10 - X Marker Pulse
Channel 11 - Y Marker Pulse
Channel 12 - Z Marker Pulse
Channel 13 - X Grid Limit
Channel 14 - Y Grid Limit

PAGE # 5 CONT.

SERIAL INPUTS (cont.)

Channel 15 - Z Grid Limit
Channel 16 - Tool Change Reset
Channel 17 - Auto Lube Fault
Channel 18 - Wait
Channel 19 - RS-232/TTY In
Channel 1A - Feed Hold
Channel 1B - Zero Speed
Channel 1C - Emergency Stop
Channel 1D - Spare
Channel 1E - Block Delete
Channel 1F - Cycle Start

2.3.6 DISPLAY STATUS

Page #6 TOOL OFFSET TABLE

There are twenty-four offsets in the system which can be displayed as a table by depressing the DISP STATUS key 6 and any number 00-24.

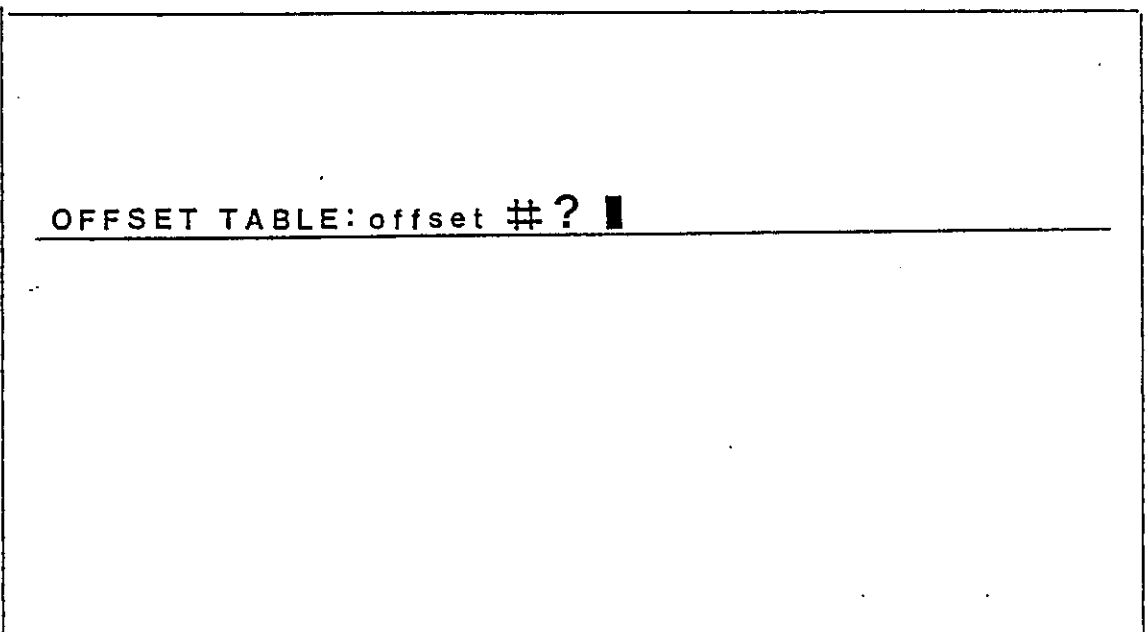


FIGURE 2.3.6 CRT DISPLAY - DISPLAY OFFSET PAGE

When an offset number (01-24) is keyed in, the CRT will display one of three tables. These are offset tables for Tool #'s 01 through 08, 09 through 16, and 17 through 24. The table displayed will contain the desired tool # and its offset values (R,X,Y, and Z). The offset number called for will be displayed in the active line of the CRT and

the operator can then enter desired values.

EXAMPLE

For Pressing

DISP
STAT

6

0

4

the following

table is displayed.

OFFSET TABLE: offset # ? 04 R

Tool # 01	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 02	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 03	R 00.0000	X 00.0030	Y 00.0000	Z 00.0000
Tool # 04	R 00.0000	X 00.0032	Y 00.0000	Z 00.0000
Tool # 05	R 00.0000	X 00.0035	Y 00.0000	Z 00.0000
Tool # 06	R 00.0000	X 00.0037	Y 00.0000	Z 00.0000
Tool # 07	R 00.0000	X 00.0040	Y 00.0000	Z 00.0000
Tool # 08	R 00.0000	X 00.0042	Y 00.0000	Z 00.0000

FIGURE 2.3.7

OFFSET TABLE 01-08

Any offset value may now be entered for offset 04.

EXAMPLE 2.3.6.1 - since the cursor is at TOOL # 04 and entry for R (radius) is up, by pressing the key sequence.

. 1 5 ENT

The following table will now be displayed with an R offset in Tool #04 of 00.1500.

OFFSET TABLE: offset # ? 04 X

Tool # 01	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 02	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 03	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 04	R 00.1500	X 00.0000	Y 00.0000	Z 00.0000
Tool # 05	R 00.0000	X 00.0035	Y 00.0000	Z 00.0000
Tool # 06	R 00.0000	X 00.0037	Y 00.0000	Z 00.0000
Tool # 07	R 00.0000	X 00.0040	Y 00.0000	Z 00.0000
Tool # 08	R 00.0000	X 00.0042	Y 00.0000	Z 00.0000

FIGURE 2.3.8 OFFSET VALUE FOR 04 ENTERED

Offset table 09-16 may be displayed by keying in one of the offset numbers.

EXAMPLE 2.3.6.2

DISP
STAT

6

1

5

will show the following table.

OFFSET TABLE: offset # ? 15 R

Tool # 09	R 00.0000	X 00.0045	Y 00.0000	Z 00.0000
Tool # 10	R 00.0000	X 00.0047	Y 00.0000	Z 00.0000
Tool # 11	R 00.0000	X 00.0048	Y 00.0000	Z 00.0000
Tool # 12	R 00.0000	X 00.0062	Y 00.0000	Z 00.0000
Tool # 13	R 00.0000	X 00.0087	Y 00.0000	Z 00.0000
Tool # 14	R 00.0000	X 00.0100	Y 00.0000	Z 00.0000
Tool # 15	R 00.0000	X 00.0112	Y 00.0000	Z 00.0000
Tool # 16	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000

FIGURE 2.3.9 OFFSET TABLE 09-16

For offsets 17 through 24 the key sequence will result in the following display.

EXAMPLE 2.3.6.3

DISP
STAT

6

1

8

OFFSET TABLE: offset # ? 18 R

Tool # 17	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 18	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 19	R 00.0000	X 00.0000	Y 00.5500	Z 00.0000
Tool # 20	R 00.0000	X 00.0000	Y 00.5500	Z 00.2500
Tool # 21	R 00.0000	X 00.0000	Y 00.5500	Z 00.2500
Tool # 22	R 00.0000	X 00.0000	Y 00.5500	Z 00.5500
Tool # 23	R 00.0000	X 00.0000	Y 00.5500	Z 01.0500
Tool # 24	R 00.0000	X 00.0000	Y 00.5500	Z 00.7000

FIGURE 2.3.10 OFFSET TABLE 17-24

It is now possible to enter values for tool offset # 18..

EXAMPLE 2.3.6.4

[ENT] [.] [5] [-] [ENT] [6] [ENT]

OFFSET TABLE: offset # ? 18 X

Tool # 17	R 00.0000	X 00.0000	Y 00.0000	Z 00.0000
Tool # 18	R 00.0000	X -04.5000	Y 00.6000	Z 00.0000
Tool # 19	R 00.0000	X 00.0000	Y 00.5500	Z 00.0000
Tool # 20	R 00.0000	X 00.0000	Y 00.5500	Z 00.2500
Tool # 21	R 00.0000	X 00.0000	Y 00.5500	Z 00.2500
Tool # 22	R 00.0000	X 00.0000	Y 00.5500	Z 00.5500
Tool # 23	R 00.0000	X 00.0000	Y 00.5500	Z 01.0500
Tool # 24	R 00.0000	X 00.0000	Y 00.5500	Z 00.7000

Figure 2.3.11 CRT DISPLAY - VALUE ENTERED FOR TOOL # 18

Note that by depressing the [ENT] key first the R offset value was bypassed and the offset values went into X and Y.

2.3.7 PROGRAM PARAMETER TABLE

KEY SEQUENCE

DISP
STAT

7

PARAMETER TABLE: parameter # ?

#00 P 00.0000	#01 P 00.0000	#02 P 00.0000	#04 P 00.0000
#04 P 00.0000	#05 P 00.0000	#06 P 00.0000	#07 P 00.0000
#08 P 00.0000	#09 P 00.0000	#10 P 00.0000	#11 P 00.0000
#12 P 00.0000	#13 P 00.0000	#14 P 00.0000	#15 P 00.0000
#16 P 00.0000	#17 P 00.0000	#18 P 00.0000	#19 P 00.0000
#20 P 00.0000	#21 P 00.0000	#22 P 00.0000	#23 P 00.0000
#24 P 00.0000	#25 P 00.0000	#26 P 00.0000	#27 P 00.0000
#28 P 00.0000	#29 P 00.0000	#30 P 00.0000	#31 P 00.0000
#32 P 00.0000	#33 P 00.0000	#34 P 00.0000	#35 P 00.0000
#36 P 00.0000	#37 P 00.0000	#38 P 00.0000	#39 P 00.0000

FIGURE 2.3.12

CRT DISPLAY OF PROGRAM PARAMETERS

This page shows the current value of all programmed parameters which can be accessed and entered directly through this page or by the G3XX code. Once a specific parameter has been called for, successive parameters can be accessed by simply hitting the [ENT] key.

2.3.8 MACHINE SETUP PARAMETERS

KEY SEQUENCE

DISP
STAT

9

CE

DISP
STAT

9

X

X

X

ENT

Where XXX is, the desired system parameter number

9 Machine Setup Mode 000 SP

000	SP	00.0000	001	SP	00.0000	002	SP	00.0000	003	SP	00.0000
004	SP	00.0000	005	SP	00.0000	006	SP	00.0000	007	SP	00.0000
008	SP	00.0000	009	SP	00.0000	010	SP	00.0000	011	SP	00.0000
012	SP	00.0000	013	SP	00.0000	014	SP	00.0000	015	SP	00.0000
016	SP	00.0000	017	SP	00.0000						

FIGURE 2.3.13 1st SET OF SYSTEM PARAMETERS IS DISPLAYED ON CRT

Shown here is the first section of system parameters. For further information see Section XV on Machine Set-up Parameters.

2.4 M-FUNCTIONS DEFINED

- M00 - PROGRAM STOP
Caused the program to stop until the cycle start button is depressed. The program stop condition will be indicated on the CRT.
- M01 - OPTIONAL STOP
Causes the program to stop whenever channel 0F is held low - typically by a selector switch. Otherwise this function is ignored.
- M02 - END OF PROGRAM
Terminates program execution, shuts off spindle, coolant, cutter comp, canned cycles, mirror image, and waits for the cycle start button to be depressed. When cycle start is pushed line 001 of the program will be executed. If the parts program is entered via the Centurion keyboard an M02 is inserted automatically at the end of each program.
- M03 - SPINDLE ON CLOCKWISE
Turns the spindle on in the clockwise direction.
- M04 - SPINDLE ON COUNTERCLOCKWISE
Turns the spindle on in the counterclockwise direction.
- M05 - SPINDLE STOP
Turns the spindle off.
- M06 - TOOL CHANGE
Turns off the spindle (Z will retract to home position on systems with tool changer option, before the spindle turn off). The CRT display will indicate that there is to be a tool change and which tool # to insert. At this point the operator should depress the TOOL CHANGE RESET pushbutton. The program will resume normal operation when the cycle start button is depressed. However, the spindle and coolant will not restart. An M06 command is normally followed by either an M03 or M04 and an M07 and M08. The M06 also causes all subroutines (except those with tool #'s specified) to be recompensated for the tool# that has been enabled prior to the M06 command.
- M07 - MIST COOLANT ON
Activates the mist coolant output.
- M08 - FLOOD COOLANT ON
Activates the flood coolant output.
- M09 - COOLANT OFF
Turns off both the flood and mist coolant outputs.
- M11-M18 - TURNS CHANNELS 1-8 ON (optional)
Unassigned M-Functions, channels 1 through 8 will be activated by M-codes M11 through M18.
- M21-M28 - TURN CHANNELS 1-8 OFF
Unassigned M-Functions, channels 1 through 8 will be deactivated by M-code M21 through M28.

M30 - REWIND TAPE

Will cause the paper tape to rewind to the first section of tape containing 16 consecutive tape feeds.

M32 - WAIT FOR CHANNEL

Upon execution of this command the control will stop if wait channel 18 is held low (contact closed) until the channel goes high (contact open).

M600 - CUTTER COMP ALL SUBROUTINES

This command can be used whenever all the existing subroutines need to be re-compensated for tool radius. This command should not be used if there is a chance that some of the subroutines might give E22-E23 (tool too large) errors because the current tool is too big. For these cases use M6XX command.

M6XX - CUTTER COMP SUBROUTINE XX

When only some of the subroutines need to be re-compensated for cutter radius the M6XX command can be used (where XX = 1 to 99 in the subroutine number). This command is very handy when the same subroutine needs to be run twice for rough & finish pass. The main program can change cutter radius with G21, G20 command and use the M6XX command and call the subroutine.

M995

This code allows the Feedrate or Spindle command which follows it to be loaded with a value from a parameter. For example,

M995 F000.9

M995 S0002

will set the feedrate equal to the contents of parameter #9, and the spindle speed to the contents of parameter #2.

For a feedrate of 45.0 IPM and a spindle speed of 3500 RPM, the program must have earlier declared:

G309 P.0450

G302 P.3500

M994

This M-function retracts Z to the tool change position. (ie: grid) This position defaults to grid position unless it is changed by parameter 079, in the machine setup tables.

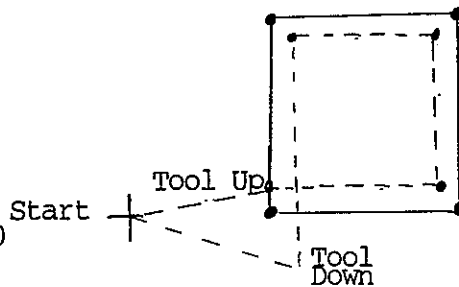
M996

The axis commands in this block will be ignored during run time. The purpose of this block is to assist cutter comp. in determining the correct compensated position for cutter entry to or exit from a part. For example, the square programmed below will be compensated as shown by the dashed line.

```

1001 X-.5 Y0
1002 G86 G42
1003 X0 Y0
1004 X0 Y1
1005 X1 Y1
1006 X1 Y0
1007 X0 Y0
1008 G40 X-.5 Y0
1009 G80 G99

```



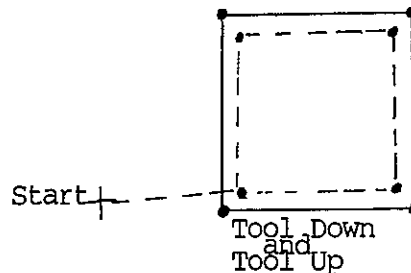
Cutter down & up at (0,0)

Using the M996 "Non-motion" block to give cutter compensation more information, the program can be rewritten to perform in a preferable manner (dotted line):

```

1001 G86 G42 M996 X1 Y0
1002 X0 Y0
1003 X0 Y1
1004 X1 Y1
1005 X1 Y0
1006 X0 Y0
1007 G40 M996 X0 Y1
1008 G80 G99

```



M997 and M998

These are used to surround a sequence of moves, M998 at the start and M997 at the end, so that the programmer can define his own pocket clearing shape. The instances in which this would aid the programmer are those requiring multiple repetitions of a sequence of cuts at increasing depths.

Parameters 6, 7, and 8 are used to indicate the first pass Z Position, the subsequent Z increment, and the final Z depth, respectively. These are set by G3XX or G4XX instructions.

When M998 is first encountered (and G86, mill, has been activated) the quill will feed to the depth which is in parameter 6. The movement described in the next blocks will be performed until the occurrence of an M997. At this point, the quill will rapid to clearance (Parameter 0 should contain the R-plane) and X and Y will rapid to the starting point. The quill will feed down to the previous depth + the increment (Parameter 7) and the blocks following the M998 block will again be executed.

This process repeats until the depth of the quill reaches the depth given in Parameter 8. Then the quill rapids to the clearance depth, and the blocks following the M997 block will be performed.

All feeds of the quill will be at the feedrate given in parameter 24, the Z-feedrate (if Parameter 24=0, at the programmed feedrate).

M999

"Block Skip" command - when the CNC executes this code it will check status of the Block Skip switch, if it is on, the CNC will skip that block of information. This command is the same as the slash code "/".

2.5 MUTUALLY EXCLUSIVE M-CODES

Although any M-Code can appear on the same program block with any other code or command, or combination of codes and commands, there are some M-Codes which would not logically appear together on the same program block. There are really two such sets of M-Codes.

- A. The first set includes the M3, 4 and 5 codes. Since the computer arranges program information so that M and G-Codes appear in a block before the axes commands, there is no logical reason for stopping and starting the spindle or for starting and reversing the spindle in a single program block.
- B. The same reasoning holds true for the second set of mutually exclusively M-Codes which includes the M7, 8 and 9 codes, or coolant commands.

2.6 G-CODES DEFINED

This section contains a list and explanation of all G-Codes. In subsequent sections of this manual some G-Codes will be dealt with again. Some functions when activated stay active through the rest of the program until they are deactivated. These functions are labeled modal.

Note: All G-Codes can be used either in MDI or in a program with the exception of 670 and G71.

G0 - RAPID TRAVERSE (modal)

This function forces the machine into operating at its rapid feedrate (typically 200 ipm). Upon initiation of this code all axis moves will take place at the rapid rate with the axis moving vectorally to the programmed point.

The programmed feed remains in the feedrate register and can be activated by cancelling the G0.

G0 cancels G1.

G1 - FEED MODE (modal)

This function cancels the G0 Rapid Traverse mode and the feedrate register becomes active. This is the feed or linear interpolation mode. In this mode the machine can be programmed to move at feeds from 0.1 ipm to 100 ipm. On power up G1 is active with feed of 20 ipm.

G1 cancels G0.

G2 - COUNTERCLOCKWISE CIRCULAR INTERPOLATION (modal)

This function results in all subsequent arcs (either programmed or part of autoroutines) to be generated in the counterclockwise direction. Other information required for arc generation is the radius of the circle, center and end point dimensions. The present machine position is assumed to be the start point. Arcs can also be generated in the XY, XZ, YZ planes. (See Section XVII.) A full 360° arc requires only 1 block.

G2 cancels G3.

G3 - XY PLANE COUNTER CLOCKWISE CIRCULAR INTERPOLATION (modal)

This function is identical to G2 except the arcs are generated in the counterclockwise direction.

G3 cancels G2.

G3 is active on power-up.

SET-UP

1. MACHINE AT X1 Y0

General Format

2. G3R1I0J0X0Y1

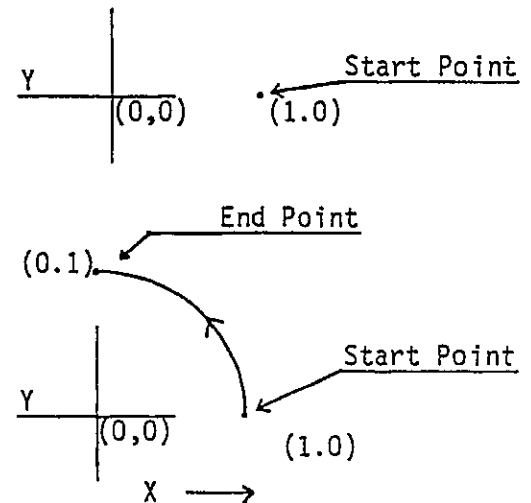


FIGURE 2.5.1 XY PLANE CCW CIRCULAR INTERPOLATION

The above will result in the generation of an arc of radius 1" from the start point with center at I0J0 to the end point at X0Y1 in the counterclockwise direction.

G4 - DWELL COMMAND

The G4 code must be immediately followed by a FXXXX.X instruction. This instruction will then cause the program to stop or dwell for XXXX.X seconds.

General Format

G4F2.5 will cause program to dwell for 2.5 seconds

G4F25 will cause program to dwell for 25 seconds.

G10 - DISPLAY Z-AXIS (modal)

This code will cause Z-axis position and floating zero location information to be displayed on the CRT whenever making axis moves or displaying DISP STAT Page 0. This code is active upon power-up and cancels the G11 code.

G11 - DISPLAY C-AXIS (modal)

This code will cause C-axis position and floating zero location information to be displayed on the CRT whenever making axis moves or displaying DISP STAT Page 0. The G11 code cancels the G10 code.

G12 - DIGITIZE MODE

This code can only be entered via the MDI mode. When a G12 FXXXX is commanded the servo drives are shut-off, so that the machine may be positioned manually, and the control accesses Block XXXX. The display now works like a digital read-out (DRO) and any X or Y axis motion will be indicated on the display. In addition, as the machine is manually guided around a part or template the part can be directly programmed by simply hitting the EOB key whenever a point is to be stored in memory. Each time the EOB key is depressed the present X and Y coordinates will be stored in the block selected and the line number will increment by one for the next entry. When the program is completed the CE key will cause the servo drives to be turned-on and the digitize mode cancelled.

General Format

G12 ENT Digitize Mode

F1001 ENT Selects program 1 line 1 (Block 1001)

Cycle Start Turns off drives and enables digitize mode.

G13 - XC PLANE CIRCLE (modal)

Selects the XC plane for any circular moves programmed (auto-routines are not affected). Other information required to generate an arc in the XC plane are: the radius of the arc, the XC center point, and the XC end point dimensions. The present machine position is assumed to be the start point of the arc.

G13 cancels G14, G15, G17, G18 and G19.

SET-UP

Assume machine at X1, C0 for the following explanation

General Format

G2,G13,R1,I0,C0,X0,C10

This will result in an arc in the clockwise direction starting at X1,C0 with center point at X0, C0 and end point at X0, C10 (C dimension are in degrees).

G2,G13,R1,I0,Y0,X0,Y-1]
G2,G13,R1,I0,Z0,X0,Z-1]

NOT VALID (Y and Z are not valid for center or end points with G13).

Note: C-axis is defined as angular motion (rotary axis around the Z-axis.

G14 - YC PLANE CIRCLE

Selects the YC plane for any circular moves programmed (auto-routines not affected). Other information required to generate an arc in the YC plane are: the radius of the arc, the YC center point, and YC end point dimensions. The present machine position is assumed to be the start point of the arc.

G14 cancels G13, G15, G17, G18 and G19.

SET-UP

Assume machine at Y1,C0 for the following explanation.

General Format

G3,G14,R1,J0,C0,Y0, C-10

This will result in an arc in the counterclockwise direction starting at Y1,C0 with center point at Y0,C0 and end point at Y0,C-10. (C dimensions are in degrees.)

G3,G14,R1,X0,Y0,X-1,Y0
G3,G14,R1,Y0,Z0,Y0,Z-1

NOT VALID (X and Z centerpoint and end point instructions are invalid with a G14 code).

G15 - ZC PLANE CIRCLE (modal)

Selects the ZC plane for any circular moves programmed (auto-routines not affected). Other information required to generate an arc in the ZC plane are: the radius of the arc, the ZC center point, and ZC end point dimensions. The present machine position is assumed to be the start point of the arc.

G15 cancels G13, G14, G17, G18 and G19.

SET-UP

Assumes machine at Z1,C0 for the following explanation

General Format

G2,G15,R1,K0,C0,Z0,C10

This will result in an arc in the clockwise direction starting at Z1,C0 with center at Z0,C0 and end point at Z0,C10 (C dimensions are in degrees).

G2,G15,X1,X0,Z0,X0,Z1
G2,G15,R1,Y0,Z0,Y0,Z1

NOT VALID (X and Y center and end point instructions are invalid with G15 code).

G17 - XY PLANE CIRCLE (modal)

Selects the XY plane for any circular moves programmed. Other information required to generate an arc in the XY plane are: the radius of the arc, the XY center point, and XY end point dimensions. The present position is assumed to be the start point of the arc.

G17 cancels G13, G14, G15, G18, G19 and is active upon power-up.

SET-UP

Assume machine at X1,Y0 for the following explanation.

General Format

G2,G17,R1,I0,J0,X0,Y-1

This results in a 90° arc in the clockwise direction starting at X1,Y0 with center at X0,Y0 and end point at X0,Y-1.

G2,G17,R1,J0,Z0,X0,Z-1
G2,G17,R1,J0,B0,Y0,B-10

NOT VALID (Z and rotary axis center and end points are invalid with G17 code.)

G18 - XZ PLANE CIRCLE (modal)

Selects the XZ plane for any circular moves programmed (autoroutines are not affected). Other information required to generate an arc are: the radius of the arc, the XZ center point, and XZ end point dimensions. The current machine position is assumed to be the start point of the arc.

G18 cancels G13, G14, G15, G17 and G19.

SET-UP

Assume machine at X1,Z0 for the following explanation.

General Format

G2,G18,R1,I0,K0,X0,Z-1

This will result in a 90° arc in the clockwise direction starting at X1,Z0 with center at X0,Z0 and end point at X0,Z-1.

G2,G18,K1,I0 J0,X0,Y-1
G2,G18,R1,I0,A0,X0,A-10

NOT VALID (Y-axis and rotary axis center and end points are invalid with G18 code).

G19 - YZ PLANE CIRCLE (modal)

Selects the YZ plane for any circular moves programmed (auto-routines not affected). Other information required to generate an arc in the YZ plane are: the radius of the arc, the YZ center point and YZ end point dimensions. The present machine position is assumed to be the start point of the arc.

G19 cancels G13, G14, G14, G17 and G18 codes.

SET-UP

Assume machine at Y1,Z0 for this explanation.

General Format

G3,G19,R1,J0,K0,Y0,Z1

This will result in a 90° arc in the counterclockwise direction starting at Y1,Z0 with center at Y0,Z0 and end point at Y1,Z1.

G3,G19,R1,I0,J0,X0,Y1
G3,G19,R1,J0,B0,Y0,B10

NOT VALID (X-axis and rotary axis center and end point instructions are invalid with G19 code).

G20 - MODIFY TOOL OFFSET

G20 is used with G1XX and X, Y, Z or R to modify (increase or decrease) existing tool offset XX. See Section VI 6.4. The R offsets must appear separately from the X, Y and Z offsets.

G21 - ENTER TOOL OFFSETS

G21 is used with G1XX and X, Y, Z or R to enter a tool offset value into tool location XX. See Section VI 6.3. The R offset must appear separately from the X, Y and Z offsets.

G26 - AUTOROUTINE FOR INSIDE/OUTSIDE POCKET FINISH

This function generates the template given in Figure 2.5.2. By pre-entering the correct values for the various parameters shown, the operator can make the desired pocket with one block of information.

Figure 2.5.2 - G26 Autoroutine

where P1 = X Axis pocket dimension

P2 = Y Axis pocket dimension

P3 - Radius of the quarter arcs on the corners

The variable parameters in Figure 2.5.2 can only have positive values.

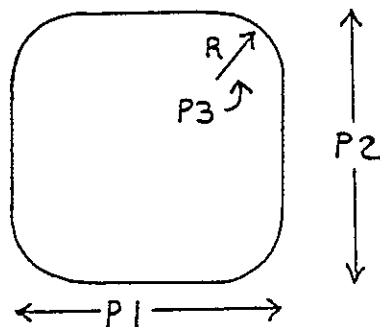


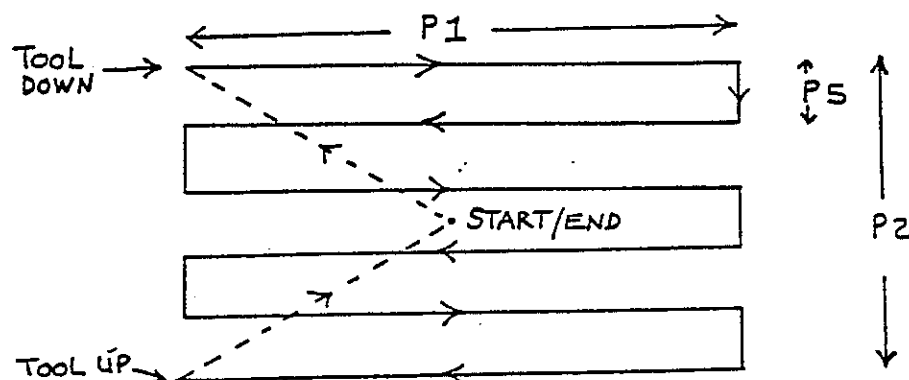
FIGURE 2.5.2 G26 VARIABLE PARAMETERS

The template in Figure 2.5.2 can be made for both inside or outside pocket finish.

The machine should be at the center of the pocket before the G26 is executed.

G26 can be used with or without cutter compensation.

This function generates the template given in Figure 2.5.3. By presenting the correct values for the various parameters shown, the operator can clear a pocket or drill symmetrical holes with one block of information.



G27 AUTOROUTINE VARIABLE PARAMETERS

P5 = Y increment to determine number of passes to clear pocket

Only inside pocket clearing can be accomplished by the G27.

For symmetric hole drilling, canned cycles (which are explained in Section VIII) can be used to automatically drill holes at the end of each linear move (except the start/end point) shown in figure 2.5.3.

Note: P6, P7 and P8 can be used in conjunction with the G86 milling canned cycle to achieve the desired pocket depth in the G27 autoroutine.

P8 = desired finish depth

If the G86 milling canned cycle is active and P6, 7 and 8 are entered prior to the G27 command, the autoroutine will run at the depth determined by the P6 on the first pass and the depth will then increase by P7 on each successive pass until the final depth specified by P8 is reached.

In addition to these parameters, parameter P4 can be given which will then cause the G27 autoroutine to leave material around the edge of the pocket (the amount of material is determined by P4) for a finish pass. The same parameter values used with the G27 autoroutine will now give an accurate finish pass

when used with the G26 autoroutine.

G28 - AUTOROUTINE FOR CIRCULAR POCKET CLEARING

This function generates the template shown in Figure 2.5.4. By pre-entering the correct values for the various parameters shown, the operator can make the desired pocket with one block of information.

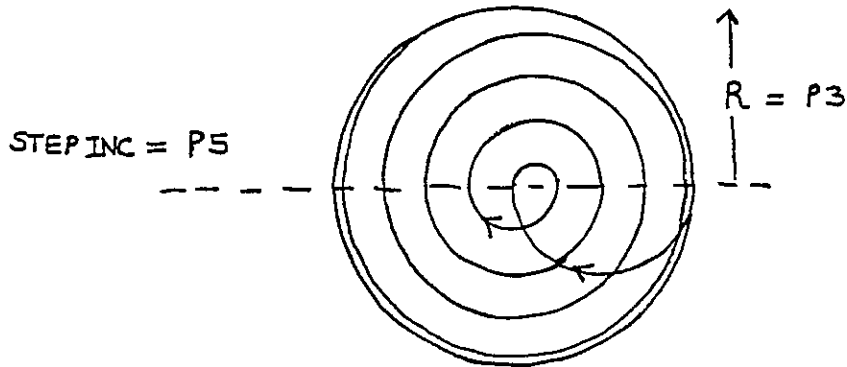


FIGURE 2.5.4 G28 AUTOROUTINE

Where P3 = Radius of the circular pocket

P5 = X increment to determine the number of passes to clear the pocket

The parameters in Figure 2.5.4 can only have positive value.

Only the inside pocket clearing can be accomplished by the G28 autoroutine. G28 can be used with or without cutter compensation.

Start and end point is at the center of this subroutine.

For using the G28 command in a program, further explanation can be found in Section IX 9.4.

Note: P6, P7 and P8 can be used with the G28 as with the G26 and G27 autoroutines.

G30 - CONTINUOUS PATH MODE (CORNER ROUNDING) (modal)

This function is used for continuous path programming where the axes are not required to stop momentarily after each move. The result is a corner rounding effect during milling operations. On power-up, G30 is active.

G30 cancels G31.

G31 - POINT TO POINT STOP MODE (modal)

This function is used for stop mode programming where the axes come to a complete stop after each move. For drilling holes, the axes will come to the exact drill location before the hole drill sequence is initiated.

G31 cancels G30.

G32 - CANCEL FLOATING ZERO

This cancels any presets mode via a G92 command.

G40 - CANCEL CUTTER COMPENSATION (modal)

This function cancels cutter compensation. After initiation of this code, the control assumes the tool radius to be zero until the compensation is turned on again. On power-up G40 is active.

A G40 cannot be used at the end of a compensated circular move.

G40 cancels G41 and G42.

G41 - TURN ON LEFT CUTTER COMPENSATION (modal)

This function turns on cutter compensation so that the cutter path will automatically be displaced to the left of the programmed path.

See Section XIII on cutter compensation.

G41 cancels G40 and G42.

G42 - TURN ON RIGHT CUTTER COMPENSATION (modal)

This function turns on cutter compensation so that the cutter path will automatically be displaced to the right of the programmed path.

See Section XIII on cutter compensation.

G42 cancels G40 and G41.

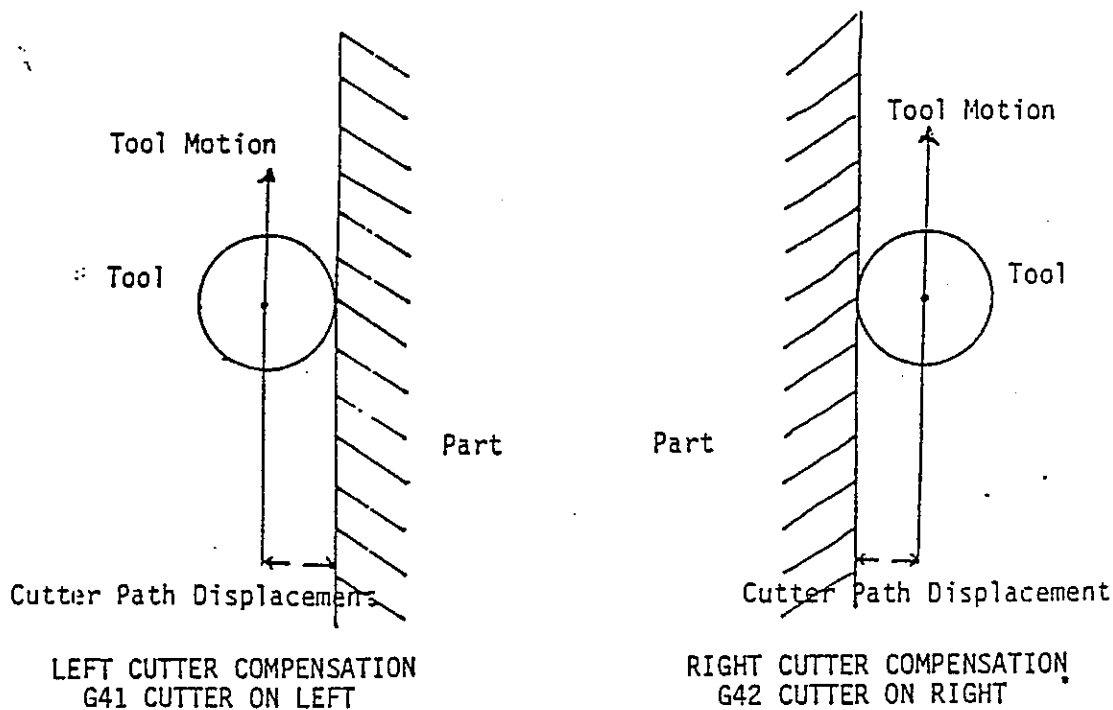


FIGURE 2.5.5 LEFT/RIGHT CUTTER COMPENSATION

DEFINITION OF LEFT/RIGHT CUTTER COMPENSATION

Figure 2.5.5 shows the difference between turning on left or right cutter compensation. The observer should look at the tool/part surface relationship facing toward the tool motion. If the tool is on the left hand side of the part surface then a displacement toward the LEFT is needed to get the tool away from the part. Thus, left cutter compensation.

Similarly, if the tool is on the right hand side of the part, then to get the tool away from the part surface it would have to be displaced to the RIGHT. Thus, right cutter compensation.

G50 - REWIND TAPE

Will cause the paper tape to rewind to the first section of tape containing 16 consecutive tape feeds.

G51 - READ PAPER TAPE

Control will "READ" the paper tape into memory. The control will accept a tape punched in either EIA or ASCII format, providing the first character is a "%".

G52 - WAIT FOR CHANNEL

Same as M32.

G53 - MEMORY TO RS-232 EIA

This function will format in EIA the parts programs and output it to the RS-232 interface.

G54 - MEMORY TO RS-232 ASCII

This function is the same as M33, except the data is formatted in ASCII.

G55 - INPUT FROM RS-232 OR TTY TO MEMORY

This function inputs from the RS-232 port, determines if the input is in EIA or ASCII and stores the data in memory.

G58 - PUNCH TAPE LEADER

This function will send out 10 inches of leader to the RS-232 or the 20 MA TTY port.

G60 - TURN MIRROR IMAGE OFF ALL AXES (modal)

This function causes cancellation of mirror image on all axes.

G60 is active with power-on.

G60 cancels G61, G62, G63, G64 and G65.

G61 - TURN ON MIRROR IMAGE X AXIS (modal)

This function will cause the X axis to be mirror imaged subsequent to its activation.

This is used for making a Left/Right hand part.

Figure 2.5.6 explains the effect of mirror imaging about X, Y axes.

G61 cancels G60.

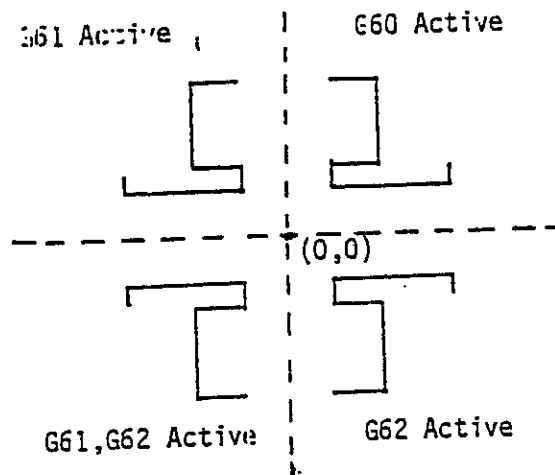


FIGURE 5.6 - EFFECT OF G61, G62

G62 - TURN ON MIRROR IMAGE Y AXIS (modal)
 This function will cause the Y axis to be mirror imaged subsequent to its activation. This is similar to G61 and is useful in making Left/Right hand parts.

Figure 2.5.6 explains the effect of mirror imaging about X, Y axes.

G62 cancels G60.

G63 - TURN ON MIRROR IMAGE Z AXIS (modal)
 This function will cause the z axis to be mirror imaged subsequent to its activation. Thus, a program to make a cavity can be inverted by activating the G63 code.

G63 cancels G60.

G64 - TURN ON MIRROR IMAGE ROTARY AXIS (modal)
 This function will cause a rotary axis to be mirror imaged (clockwise command will cause counterclockwise rotation and vice versa) subsequent to its activation.

G65 - AXIS INTERCHANGE (modal)
 This code will cause X and Y commands to be interchanged.

These codes (G60, G61, G62, G65) can be used to mirror image parts and to rotate the coordinate system in 90° intervals. The codes can be used singly or in combination. The following illustrations depict the various coordinate changes that can be achieved.

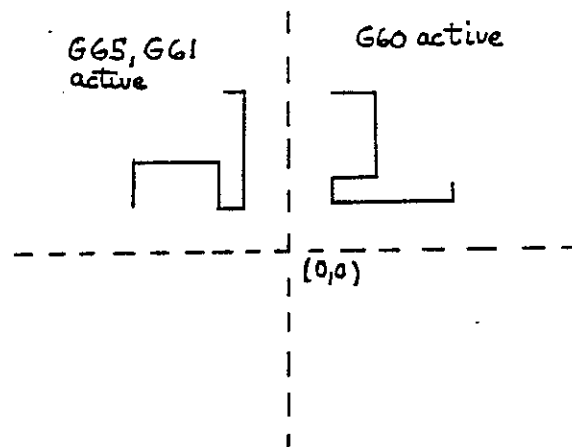


FIGURE 2.5.7 - PART ROTATION USING MIRROR IMAGE AND AXIS INTERCHANGE

Note: Care must be taken when using these codes. They are modal and will remain in effect until cancelled with G60.

The combinations depicted in drawings #2, 3, 4 and 8 will yield a generally counterclockwise direction when the original part is programmed in a clockwise direction. If cutter compensation is being employed, it is necessary to also change the DIRECTION of cutter compensation. (ie: G41 to G42).

G70 - INCH DIMENSIONING MODE (modal)

This function will cause the system to go into the Inch mode. In this mode, the system will accept dimensions in inches. This function cannot be initiated in a program; it can only be initiated in MDI.

G70 is active after power-up.

G70 cancels G71.

G71 - METRIC DIMENSIONING MODE (modal)

This function will cause the system to go into the metric mode. In this mode, the system will accept dimensions in millimeters (mm). This function cannot be initiated in MDI. In metric, the actual machine position may not exactly agree with the program position, because of the conversion. Feedrate in the metric mode is in millimeters per minute (mm/min).

G79 - RE-ENABLE CANNED CYCLE (modal)

This function is used to re-enable canned cycles, G81-G89. Canned cycles are active upon command but can be disabled or turned off with a G80 command. The last active canned cycle then be re-enabled by the G79 command.

General Format

G89 - This will cause the previously active canned cycle to become active again.

G80 - TURN OFF CANNED CYCLES (modal)

This function turns off any active canned cycles. The canned cycles remain in memory and can be reactivated again by using a G79 command.

G80 is active on power-up.

G80 cancels G81, G82, G83, G84, G85, G86, G87, G89, and G79.

Note: Canned cycles G81-G89 become active upon the completion of the first axis move after they are commanded.

G81 - DRILL CYCLE (modal)

A G81 followed by a drill depth Z dimension will cause the Z-axis, upon the completion of any axis move, to do the following.

1. Rapid to R-plane
2. Feed to Z depth
3. Rapid back to R-plane

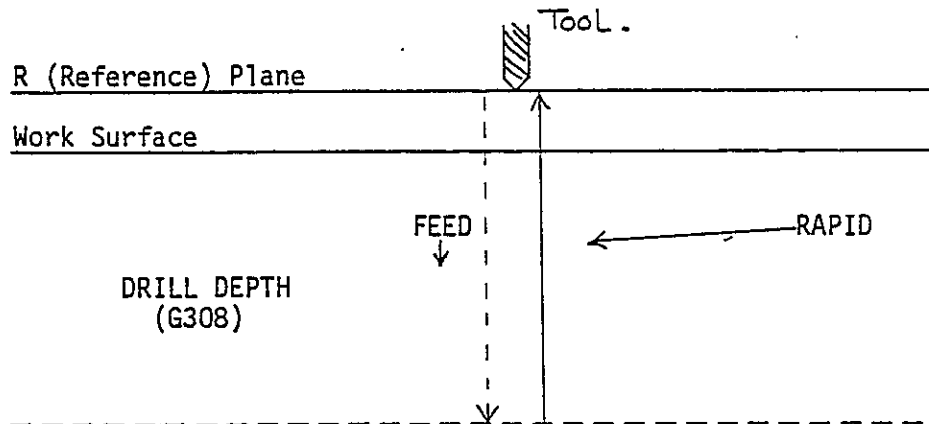


FIGURE 2.5.8 DRILL CYCLE

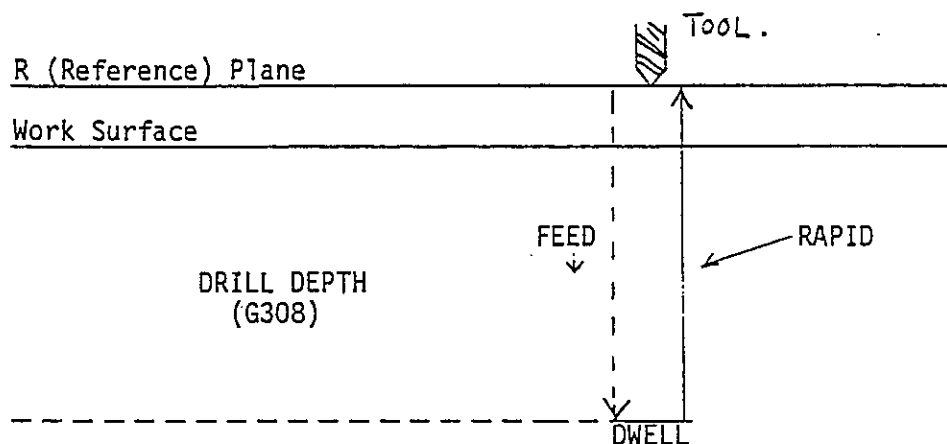
G 81 cancels G80, G82, G83, G84, G85, G86, G87 and G89

Note: R-plane is a reference or clearance plane, just above the work surface, where there are no obstructions allowing rapid moves without fear of hitting fixtures on the part piece.

G82 - DRILL WITH DWELL (modal)

A G82 followed by a Z drill depth dimension will set up a canned cycle which upon the completion of any axis move will do the following.

G82 cancels G80, G81, G83, G84, G85, G86, G87 and G89.



Note: The dwell time is set with the G323 code followed by a PXX.X where XX.X is the time in seconds.

FIGURE 2.5.9 DRILL WITH DWELL CYCLE

General Format

F20,G0		This will set-up a canned cycle for drilling
G300	P.1	with programmed dwell (.1 to 99.9 seconds).
G308	P-1.5	Machine will position to XY dimension, Z
G323	P2.0	will rapid to the R-plane (G300), then feed
G82		down to Z-1.5, dwell for 2 seconds (G323) and
		rapid back to the R-plane.

G83 - PECK AND DRILL CYCLE I (modal)

A G83 canned cycle, followed by a Z depth dimension, upon the completion of any axis move, will cause the Z-axis to do the following.

G83 cancels G80, G81, G82, G84, G85, G86, G87 and G89.

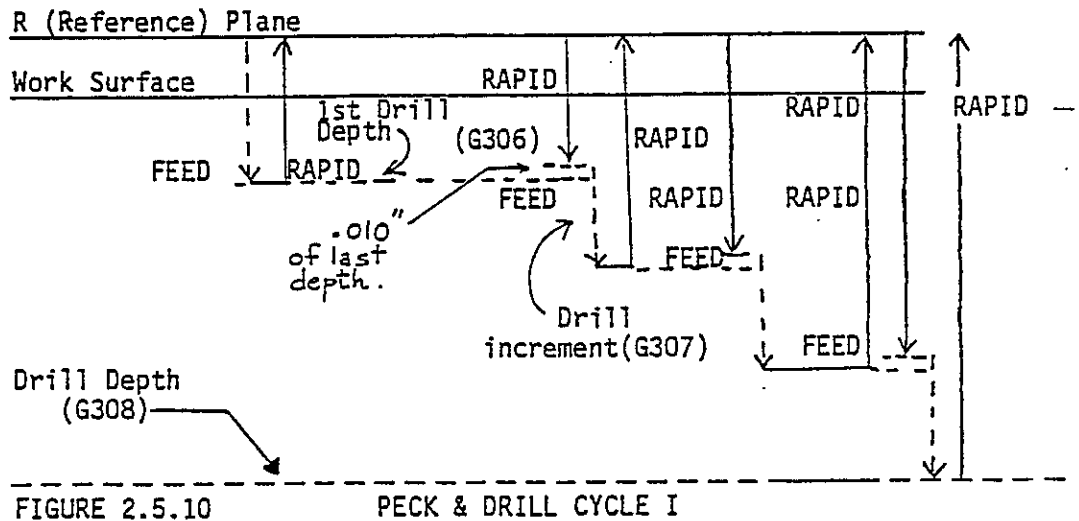


FIGURE 2.5.10

PECK & DRILL CYCLE I

Note: G308, .PXX.XXXX will enter the drill depth into parameter #8 or the canned cycle code can be immediately followed by a P dimension which will go into parameter #8.

G306, PXX.XXXX will put the first depth into parameter #6. This should be a signed number just like the final Z depth (parameter #8).

G307, PXX.XXXX will put subsequent increments into parameter P7. This is an unsigned number.

General Format

F20,G0		This will set-up a canned cycle for Peck
G300	P.1	and Drill. The machine will position to
G306	P-.2	an XY dimension. Z will rapid to the R-
G307	P.15	plane (G300) and then feed to the Z first
G308	P-1.5	depth (G306). After this initial feed,
G83		Z will rapid up to the R-plane. Then Z
		will rapid to within .010" of the previous
		feed depth.

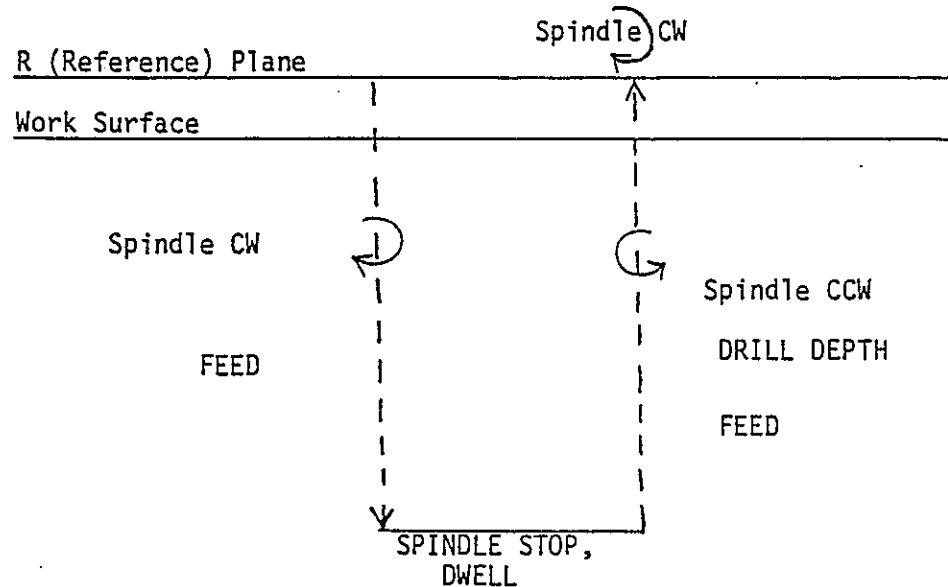
From now on, after each move, the Z-axis will feed down one increment (G307) plus .010", then rapid to the R-plane. This will continue until the Z depth (G308) is reached.

NOTE: Parameters 6 and 8 are signed values while parameter 7 is unsigned with relation to the tool length zero setting.

G84 - TAPPING CYCLE (modal)

This function will set up a canned cycle for tapping, which whenever enabled will do the following.

G84 cancels G80, G81, G82, G83, G85, G86, G87 and G89.



Operator make sure spindle is running clockwise for 1st hole.

FIGURE 2.5.11 TAPPING CYCLE

General Format

F20,G0

G300 P.1

G308 P1.5

G323 P.5

G84

After each move to an XY demension this cycle will cause the Z-axis to feed down to the Z depth (G308) where the spindle will stop. The control will now dwell for .5 seconds (G323), then the spindle will start in the counterclockwise direction, and Z will simultaneously feed back to the R-plane. The spindle will then reverse direction.

G85 - BORE CANNED CYCLE (modal)

A G85 canned cycle upon the completion of any axis move, will do the following.

G85 cancels G80, G81, G82, G83, G84, G86, G87 and G89.

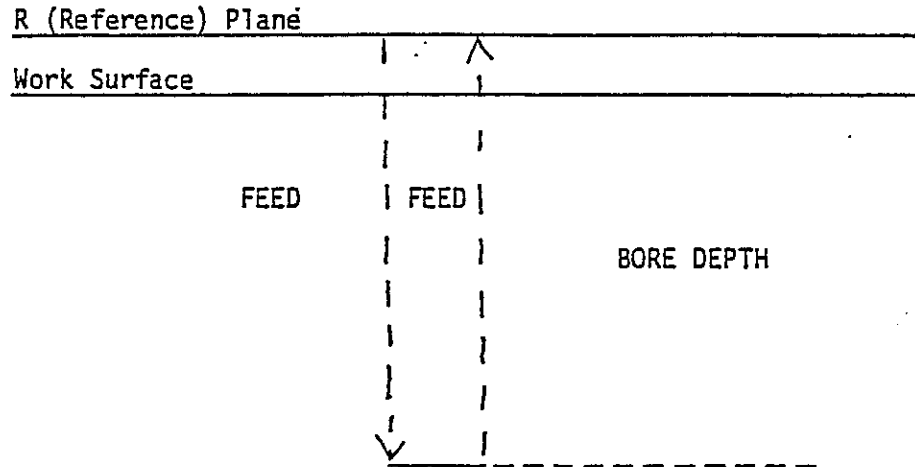


FIGURE 2.5.12 BORE CYCLE

General Format

F20G0G300P.1
G85Z1.5

This cycle will cause a hole to be bored after each XY move. Z will feed (F20) to depth (1.5" set with G85 code), then feed to the R-Plane (G300).

G86 - MILL CANNED CYCLE (modal)

A G86 followed by a Z depth dimension will set up a canned cycle which whenever enabled will do the following.

G86 cancels G80, G81, G82, G83, G84, G85, G0, G87 and G89.

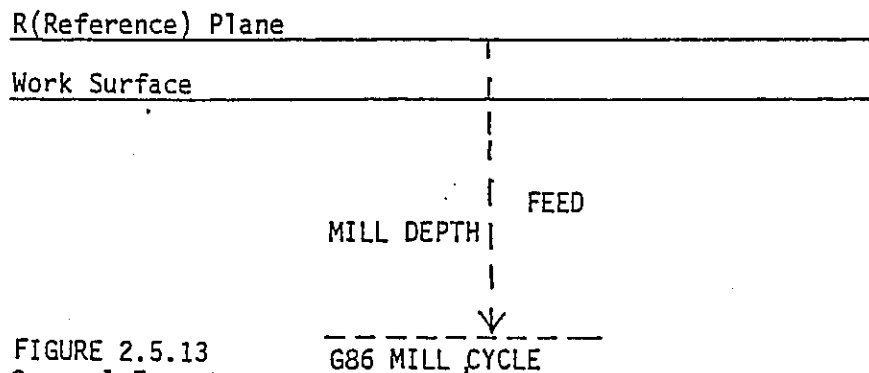


FIGURE 2.5.13
General Format

F20 G0G300P.1
G86Z1.5

This sets-up the milling cycle. After an initial XY move Z will feed to depth (1.5" set with G86 code) and remain there until the G86 is shut off.

G87 - PECK AND DRILL CYCLE II (modal)

The G87 canned cycle, upon completion of any axis move, will cause the Z-axis to do the following.

G87 cancels G80, G81, G82, G83, G84, G85, G86 and G89.

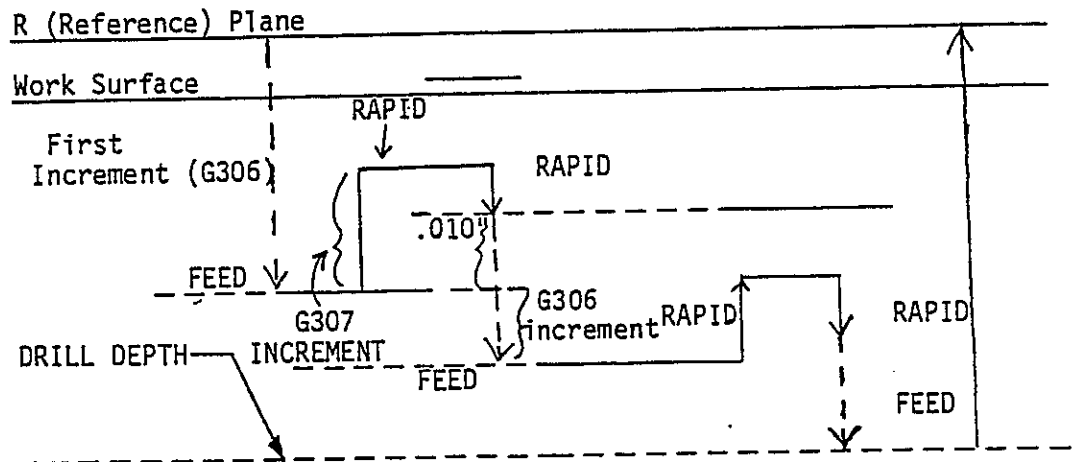


FIGURE 2.5.14 PECK & DRILL CYCLE II

NOTE: The difference between G87 and G83 is that the G87 canned cycle does not rapid to the R-plane after each feed increment but to a point equal to P7 (subsequent depth parameter) above the current depth. Then rapid to within .010" above the current depth.

G89 - BORE WITH DWELL CYCLE (modal)

This is similar to G85 except at the bottom bore depth, the control will dwell for the dwell time established by the G323, PXX.X code where PXX.X is the dwell time in seconds.

G90 - ABSOLUTE MODE (modal)

This function causes the control to go into its normal operating Absolute mode. In this mode, all dimensions are referenced from a single reference point. This reference can either be the "Absolute Zero" point which is a fixed point on the machine, or an operator defined "Floating Zero" point, which can be anywhere within the physical dimensions of the machine.

X Y Z dimensions relative to the "Absolute Zero" can only be positive because of the table/saddle limitation. Dimensions relative to "Floating Zero" can be either positive/negative or only positive, depending on where the operator sets the "Floating Zero".

G90 is active on power-up.

G90 cancels G91.

G91 - INCREMENTAL MODE (modal)

This function causes the control to go into the incremental mode. In this mode, all dimensions are entered relative to the machine position in the previous block. In the case of MDI, the dimensions are relative to the current machine position. Dimensions in G91 can be either positive or negative. Care should be taken in using G91. Whenever activating tool offsets, R-Plane dimensions or setting "Floating Zeroes" via G92, the control should not be in the G91 mode.

G91 cancels G90.

G92 - PRESET X, Y, Z, C PRESENT POSITION

This function enables setting a Floating Zero in any axes. The present position can be set to any desired value by following a G92 by the desired preset in X,Y,Z,C axis. The control then automatically sets the proper floating zeroes.

General Format

G92 XOYOZO

This will cause the current position register to show "00" in all axes. However, the machine position relative to Absolute Zero is still retained in the control.

For more detail, See Section IV 4.2

G98 - RETURN TO FLOATING ZERO

This function will cause the axes to rapid back to the last Floating Zero position. Z-axis will rapid to zero before X and Y. If no Floating Zero was set, the axis will return to the Absolute Zero (grid zero or home) position.

G99 - RAPID TO R-PLANE

This function causes Z-axis to rapid to the present R-Plane.

G100 - DEACTIVATE TOOL OFFSETS (modal)

This function will cause X, Y and Z tool offsets to be deactivated from the active tool offset registers. However, the offsets still remain intact in the memory.

G100 deactivates G1XX (XX = 01 through 24).

G100 does not effect G40, G41 or G42 radius compensation.

G1XX - ACTIVATE TOOL XX (XX = 01 through 24) (modal)

This function will cause XX to become the current tool and its X, Y and Z offsets will be activated. The range of the tool number is 01 - 24.

General Format

G101

This will make Tool #1 the current tool and activate its X, Y and Z offsets.

Also, when cutter compensation is activated, the radius value of Tool #1 will be used.

G3XX - USED TO ENTER PARAMETERS

The following list describes the uses of the different parameters.

G300 = Parameter 0 (P0)	R-Plane Value
G301 = Parameter 1 (P1)	Length of X side
G302 = Parameter 2 (P2)	Length of Y side
G303 = Parameter 3 (P3)	Radius of rounded corners, or circular pocket, or bolt hole routine.
G304 = Parameter 4 (P4)	Amount of stock to leave for a finish pass.
G305 = Parameter 5 (P5)	X or Y increment for pocket clearing
G306 = Parameter 6 (P6)	Z-Axis 1st pass depth (signed)
G307 = Parameter 7 (P7)	Z-Axis Increment (unsigned)
G308 = Parameter 8 (P8)	Z-Axis Final Depth (signed)
G309 = Parameter 9 (P9)	Z-Axis Finish Stock
G310 = Parameter 10 (P10)	Bolt Hole first angle/10
G311 = Parameter 11 (P11)	Bolt Hole = # of Holes to be drilled
G312 = Parameter 12 (P12)	Are used by the Autoroutines.
to	
G322 = Parameter 22 (P22)	
G323 = Parameter 23 (P23)	Dwell for Canned Cycle
G324 = Parameter 24 (P24)	Special feedrate for Z-axis in Canned Cycles and Auto- routines (if P24=0 programmed feed is used)
G325 = Parameter 25 (P25)	Next position for "B" axis } used in 6
G326 = Parameter 26 (P26)	Next position for "C" axis } axis system
G384 = Parameter 84 (P84)	X scale factor
G385 = Parameter 85 (P85)	Y scale factor
G386 = Parameter 86 (P86)	Z scale factor
G387 = Parameter 87 (P87)	A scale factor
G388 = Parameter 88 (P88)	S1 scale factor
G389 = Parameter 89 (P89)	S2 scale factor
G393 = Parameter 93 (P93)	Rotary angle
G371 = Parameter 71 (P71)	
to	
G399 = Parameter 99 (P99)	Are used by the Autoroutines

Note: The Z/K key is used to enter parameter values following the G3XX or G4XX codes.

G4XX - USED TO MODIFY PARAMETERS

This code is used to increase or decrease parameter values entered via the G3XX code.

G7XX - SUBROUTINE DECLARATION

See Section X

G8XX - "GOSUB" COMMAND

See Section X

G8XX - LYY LOOP ON SUBROUTINE COMMAND

See Section X

G9XX - BOLT CIRCLE COMMAND See Section IX 9.5

2.7 MUTUALLY EXCLUSIVE G-CODES

This refers to G-codes which cannot appear on the same program line with each other. The rule that governs this is as follows:

Any G-code which must be followed by axis information must not appear in the same program line with another of these type codes or with any axis move commands.

This applies to the following G-codes: G20, G21, G92, G3XX, G4XX. These are the G-codes which cannot coexist on the same program line nor can they appear with X, Y or Z move commands. In addition to these codes the G7XX code for declaring a subroutine and for returning from a subroutine must not appear on a program line with any other information or commands.

SECTION III

MACHINE START-UP PROCEDURE

3.1 MACHINE START-UP AFTER AN ELECTRICAL SHUTDOWN

1. Turn ON the MAIN DISCONNECT switch which will supply power to the control.
2. If the control is ready for operation, the console will display an E03 or E07. If an E03 or E07 does not appear, turn off power and wait 15 seconds before turning power back ON. An E03 or E07 should now be displayed.
3. Check machine travels to insure that all axes are not on or beyond a limit switch.
4. Now press the EMERGENCY STOP RESET pushbutton to reactivate the drive motors.
5. The machine is now ready to accept a GRID (Machine Reference or Home) command. See Section IV 4.1.
6. Manually move the slides to provide a well lubricated slide surface. Slides should be moved entire travel of machine.
7. Perform the setting of Floating Zero. See Section IV 4.2.
8. Perform tool reference if necessary. See Section VI 6.2.
9. Select the desired mode for operation.

3.2 MACHINE START-UP FROM NORMAL SHUTDOWN

1. Check machine travels to insure that all axes are not on or beyond a limit switch.
2. Now press the EMERGENCY STOP RESET pushbutton to reactivate the drive motors.
3. The machine is now ready to accept a GRID (Machine Reference) command (if necessary). See Section IV 4.1.
4. Manually move the slides to provide a well lubricated slide surface. Slides should be moved entire travel of machine.
5. Perform the setting of Floating Zero (if necessary). See Section IV 4.2.

6. Perform tool reference (if necessary). See Section VI 6.2.
7. Select the desired mode for operation.

SECTION IV

BASIC CONTROL OPERATION

4.1 GRID (MACHINE REFERENCE)

The machine must accept a GRID (Machine Reference) before a floating zero can be set. The GRID reference or "HOME" position is the location to which the control will reference all positions, and it must be set before any axes moves are attempted. Once the GRID reference is set, the machine is at its X, Y, and Z reference dimensions. All axes will position automatically to their referenced position at the rapid feedrate and referenced positions entered into the control if the following procedure is followed.

(Step #1 not necessary if an E03 was displayed on "POWER ON")

1. If the machine displayed an E07 on "POWER ON", the system parameter tape must be entered before the machine will accept a GRID command. To enter the System Parameter Tape:

A. Place the System Parameter Tape in the Tape Drive

B. KEY SEQUENCE: READ TAPE 9 8 ENT

C. SYSTEM PARAMETER TAPE may be removed at any time after E03 is displayed.

2. To GRID the machine:

A. Press : GRID CYCLE START

Machine will automatically GRID in all axes at the same time.

Note: When in GRID CYCLE, the FEEDRATE OVERRIDE switch has no effect. The GRID CYCLE will be executed even though the FEEDRATE OVERRIDE switch is in the FEED HOLD position.

CYCLE STOP Pushbutton will stop motion.

4.2 SETTING X, Y FLOATING ZERO "0" (PART REFERENCE)

The FLOATING ZERO position can be set anywhere within the machine travels and is always referenced to "Absolute 0" position. The location of this point is usually determined by the way the print is dimensioned, or by the design of the holding fixture.

G92 will set the FLOATING ZERO to the commanded X, Y and Z positions.

4.2.1 METHODS OF FINDING AN X, Y FLOATING ZERO "0"

1. Using an "edge finder" to locate the edges of a workpiece. To start the spindle, press Spindle CW or Spindle CCW. The spindle can be controlled by the Spindle CW, CCW and STOP switches on the operator's panel. see diagram 4.2.1

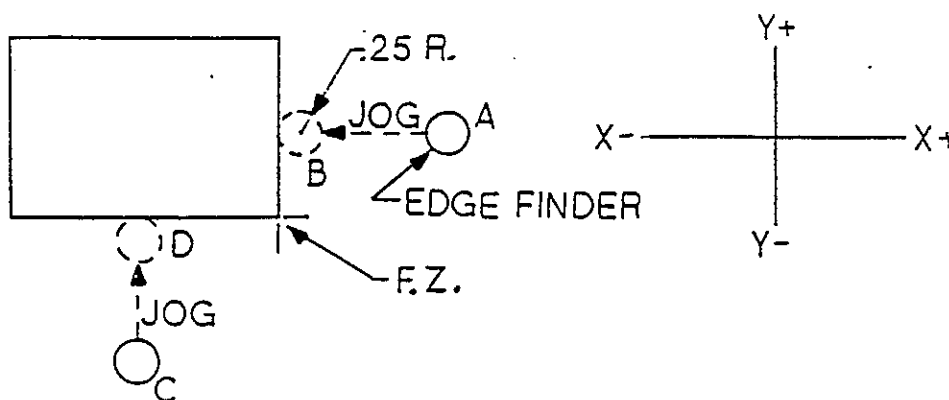
By using a slow feedrate and JOG incremental, find the edge of the workpiece or fixture. However, this position is not the center of the tool. This makes it necessary to add or subtract the radius of the "edge finder" to the position. This can be done by either:

- A. Physically moving the machine the radius and
Enter via MDI: G92 X0 (or Y0), depending on the axis or
- B. Establish if the "edge finder" is plus or minus of the desired FLOATING ZERO. If plus, enter G92 X+.RRRR (or Y+.RRRR), where, .RRRR is the radius of the "edge finder" and if minus, enter G92 X-.RRRR (or Y-.RRRR).

Note: When entering G92 (FLOATING ZERO) always make sure there is no tool offset active.

Also remember to press CYCLE START or the G92 will not be transferred to the active registers.

4.2.1 Example for setting Floating Zero using edge finder.



2. Another method can be employed when using a dial indicator to locate the center of pins or holes. Using either incremental JOG, MDI or handwheel to position the indicator to the center of the pin or holes. A G92X0Y0 can be used to set the X and Y FLOATING ZEROES.

(eg.

MDI	G	9	2	ENT	X	0	ENT	Y	0	ENT	Cycle Start
-----	---	---	---	-----	---	---	-----	---	---	-----	-------------

).

FOR X AT
POSITION B:

MDI
G
9
2
ENTER
X
.
2
5
ENTER
CYCLE START

FOR Y AT
POSITION D:

MDI
G
9
2
ENTER
Y
-
.
2
5
ENTER
CYCLE START

3. The Handwheel can also be used much like incremental JOG was used in method 1 above to locate a Floating Zero. Remember that when using the handwheel the feedrate override switch will control the sensitivity of the handwheel to allow very fine axes moves.

Individual preference will determine which of the above methods will be used to set the FLOATING ZEROES.

4.3 JOG MODE

JOG mode is activated by pressing the JOG pushbutton. The EMERGENCY STOP RESET pushbutton must be activated prior to selecting the JOG mode. Some of the functions which can be performed in JOG mode are:

1. Axis Feed Motion
2. Incremental Feed
3. Locating Floating Zero (See Section IV 4.2.1)

4.3.1 JOG PUSHBUTTON (Located on Keypad)

PRESS [JOG]

THE NUMERIC KEYPAD NOW
BECOMES A "JOYSTICK"

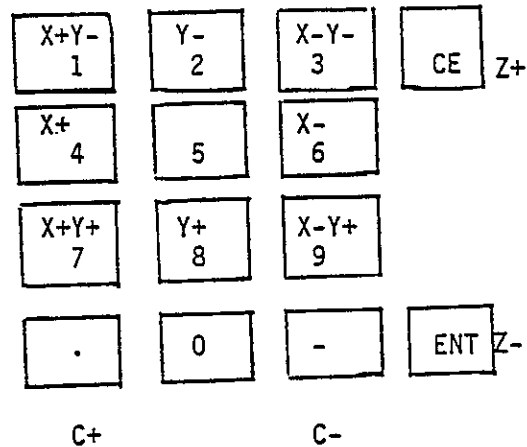


FIGURE 4.3.1 - NUMERIC KEYPAD AS JOG DIRECTION COMMAND CONTROL

4.3.2 FEED SELECTION

When feedrate switch is between 10 - 150% the machine will move as long as a key is pushed down. Incremental JOG becomes active when the switch is moved to the .1, .01, .001, .0001 positions and a key is pressed for direction and axis.

4.3.3 INCREMENTAL FEED SELECTION

In the JOG mode, the axis may be moved short, precise distances by using the INCREMENTAL FEED. The increments of movement are .0001, .001, .01 and .1. The feedrate cannot be varied. To use the INCREMENTAL FEED mode, select the desired increment of movement on the FEEDRATE OVERRIDE SELECTOR switch. Press the desired AXIS and DIRECTION SELECTOR key to initiate the incremental movement. Motion will stop when the machine slide has completed the selected increment. Pressing the SELECTOR switch, after the completion of the slide movement will cause the slide to move another increment. Pressing the push-button during movement will not cause additional slide movement.

Notes:

1. Remember to reset machine before selecting the JOG mode.
2. Select the proper Feedrate Override switch setting. Axis motion will not occur if this switch is in the feed hold position.
3. The software travel limits will not allow an axis move command to exceed the axis travel limits. Therefore, if an overtravel limit is exceeded it may be due to a control or servo loop malfunction and caution should be used when resetting the machine.

4.4 ELECTRONIC HANDWHEEL

The Centurion IV can also be operated via the electronic handwheel. Furthermore, the control display can be used as a DRO when operating the machine in this mode. In addition a part can be digitized or programmed directly when using the electronic handwheel. To center a program using this mode of operation, use the following sequence:

```
DISP PROG ----- Mode
1 -----Program Number
0
0 ----- Line Number of Program
1

HDW ----- Handwheel and axis
X, Y, Z or C
```

The selected axis (X) can now be positioned to any location using the electronic handwheel. When the ENT key is depressed, the present selected axis (X) position will be entered into Block 1, Program 1 and the Y axis will automatically be enabled or if Y had been selected Z will be enable. Another axis (Z) may be selected by pressing HDW pushbutton and the desired axis pushbutton. The selected axis (Z) may now be positioned using the electronic handwheel.

After all axes information has been entered, G, M, S, F codes may be entered.

When all block information has been entered, press the EOB pushbutton. By pressing EOB, all block information will be stored and the next block will be scrolled up to accept data either by manually loading (See Section V5.2) or again by using the electronic handwheel.

To use the electronic handwheel, press HDW pushbutton, and then the desired axis pushbutton.

General Format

[HDW]	Selects the Handwheel mode of operation
[X]	Selects the X-axis so the Electronic Handwheel will now control the X-axis.

Notes:

1. When in the Handwheel mode the Feedrate Override switch controls the sensitivity of the electronic handwheel. At the lower % settings of this switch the axes can be moved in very fine increments with the handwheel. This is to facilitate locating Floating Zeroes and setting tool length offsets. At the higher % settings of the Feedrate Override switch the axes can be moved, more rapidly with the handwheel.

2. After selecting the Handwheel mode, the 1, 2 and 4 keys select the display magnification. The 1 key selects the normal display size which is used to display the maximum amount of information. The 2 key doubles the size of the display for easier reading. The 4 key again doubles the size of the display for easiest reading and causes only current position information to be displayed. The magnification keys are also operative in MDI or while running a program but only when the axes are in motion.

4.5 MDI GENERAL

MDI is used to enter entire blocks of data by using the KEYBOARD. This allows the operator to do any machine command. MDI stands for Manual Data Input.

All that is necessary to enter the MDI mode is to push the MDI key. Information entered in the MDI mode is not stored in memory and therefore can only be used once unless it is re-entered through the KEYBOARD.

EXAMPLE 4.5.1 - MDI SINGLE ENTRY SEQUENCE

Key Sequence:

MDI ----- Mode Select
X ----- Axis Selection
1
0 ----- Dimension (10 inches)
ENT ----- Loads Dimensions into Control
CYCLE START ----- Machine moves to 10" in X

Note: Once MDI mode is selected, control will stay in this mode until another mode is activated.

EXAMPLE 4.5.2 - MDI MULTI-ENTRY SEQUENCE

Multi-function MDI Key Sequence:

MDI ----- Mode
G
0 ----- Rapid Feed
ENT ----- Command
X
1
5
ENT -----X 15"

Y

1

0

ENT ----- Y 10"

Z

-

1

ENT ----- Z-1

CYCLE START ----- Commands entry execution causing
a rapid move in X to 15", in Y
to 10", in Z to -1".

4.5.1 DO'S AND DON'TS OF MDI MODE

A. DO'S

1. Remember to RESET machine
2. Have desired FEEDRATE INSERTED
3. Have FEEDRATE OVERRIDE switch OUT OF HOLD position.

B. DON'TS

1. Program an Arc move without MDIing to the arc starting point.
2. Attempt to use Cutter Compensation or Subroutines
3. Attempt to use MDI off an axis overtravel switch

SECTION V

DISPLAY AND ENTER PROGRAMS

5.1 DISPLAY AND ENTER PROGRAMS

The Centurion IV control parts program memory has a standard capacity of approximately 10K BYTE, which can be divided into nine segments. Eight of these are used for parts programs and the ninth is used for a maximum of 99 subroutines.

5.2 PROGRAM ENTRY

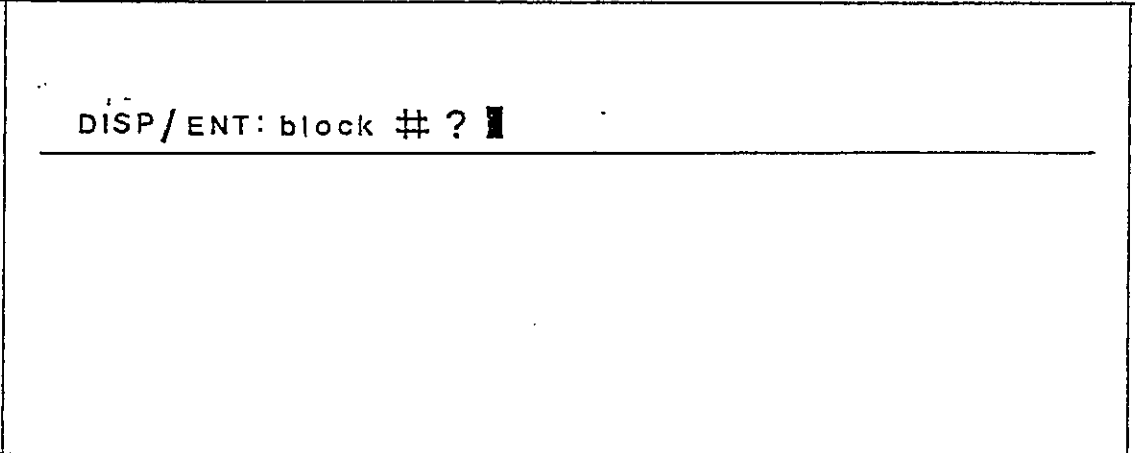
Although not necessary, it is recommended when entering programs to have the program written in detail prior to entering it into the control. This practice will minimize entry errors. To enter programs the edit key switch must be in the ON position.

EXAMPLE

1001 X1.0 EOB

This sample program consists of 1 line containing an X-axis move of 1". The following steps are required to enter it into memory.

DISP PROG -----Mode select (This key is used to both display and enter programs).



DISP/ENT: block # ? █

FIGURE 5.2.1

CRT DISPLAY - DISPLAY/ENTER PROGRAM, STEP 1

1 ----- Calls for program #1 to be displayed

0

0 ----- 3 digit line number will cause 1st line of program 1 to be called-up and displayed. If there is no information in line 1 the cursor will drop down to the line below the DISP/ENTER line.

1

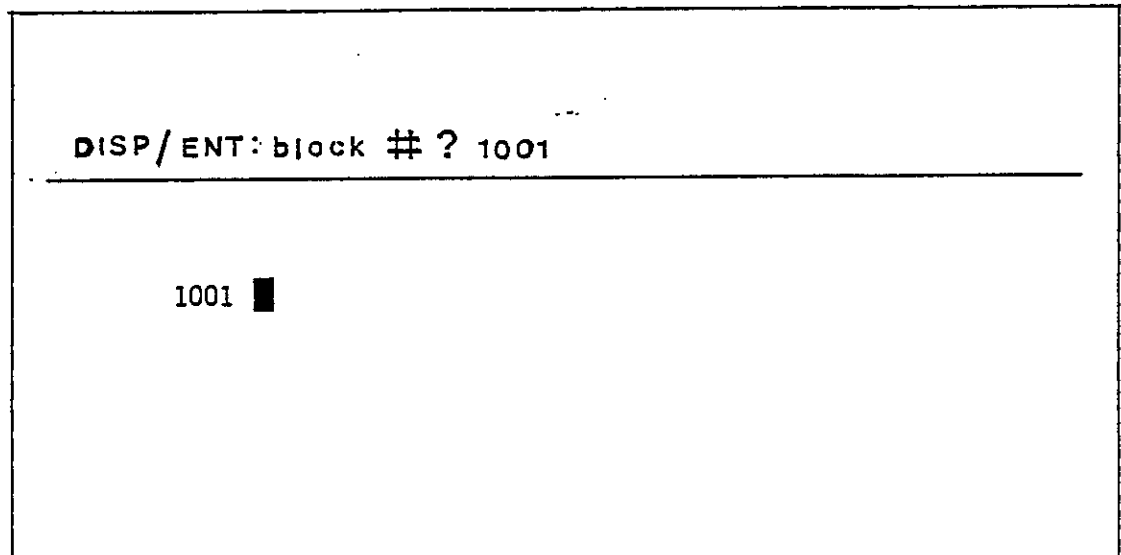


FIGURE 5.2.2 CRT DISPLAY - ENTERING PROGRAM, STEP 2

Now the control is ready to accept the first command for line 1 of the program.

X ----- X axis instruction command

1 ----- Control will see this as a dimension of 1"

As an instruction is entered it will be displayed on the DISP/ENTER line of the CRT. It is possible at this point to change the instruction number by depressing the CE key. To change the instruction code, simply select the instruction desired.

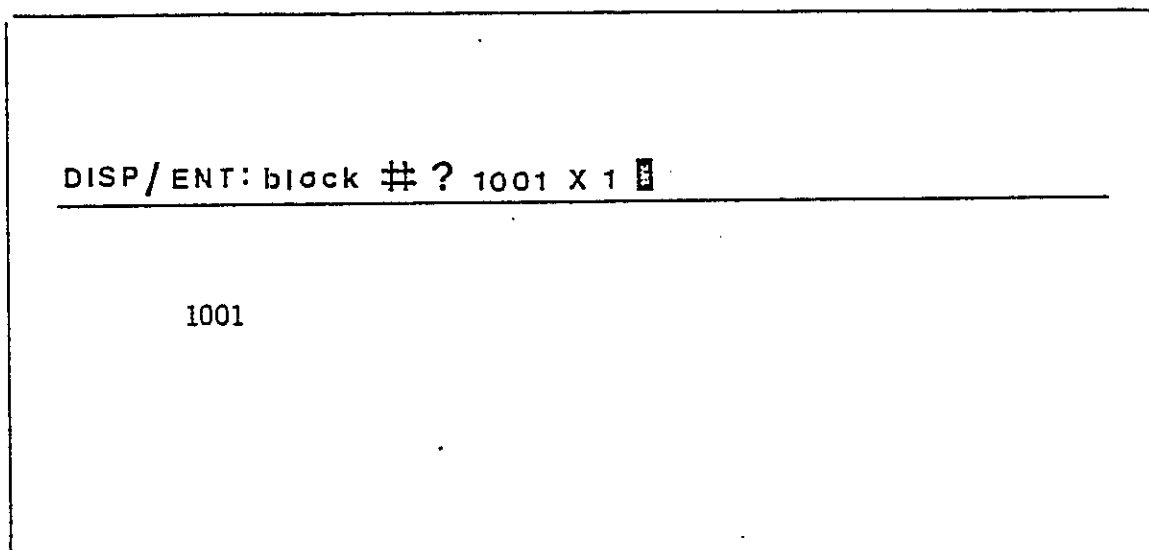


FIGURE 5.2.3 CRT DISPLAY - ENTERING PROGRAM, STEP 3

ENT ----- Causes the X1" instruction to be accepted.

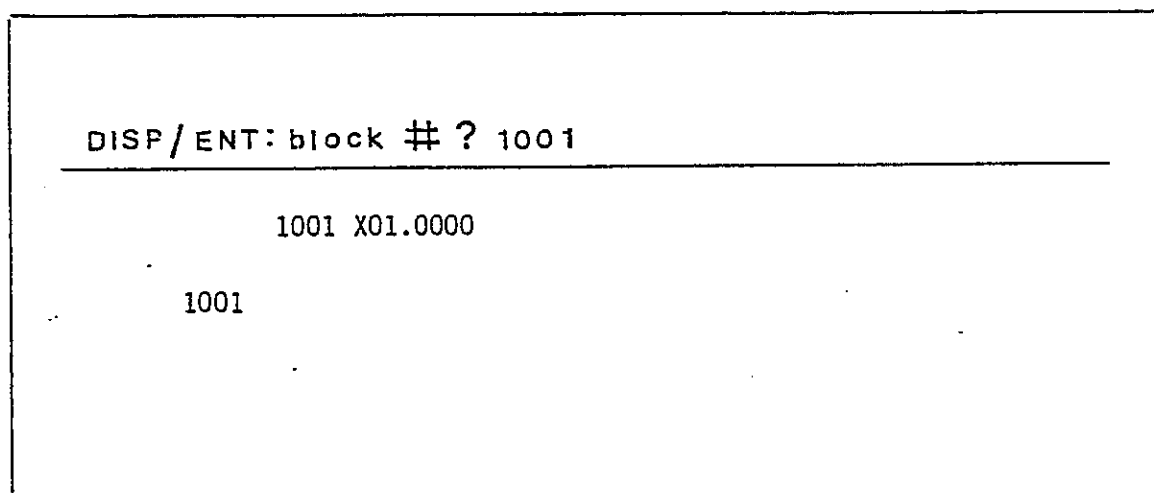


FIGURE 5.2.4 CRT DISPLAY - ENTERING PROGRAM, STEP 4

When the instruction code is complete the ENT key is depressed. The instruction is now displayed immediately below the DISP/ENTER line. Additional instructions may be entered into the line in the same way the X1" instruction was entered. To change any instruction at this point requires re-selecting the DISP PROG key and line number and re-entering all the information. Multiple M and G codes are allowed on a single line; however, they should not conflict with each other (See statements on mutually exclusive M and G codes. Sections 2.5 and 2.7).

EOB ----- This will store the entire line
into memory and advance the
Block Number

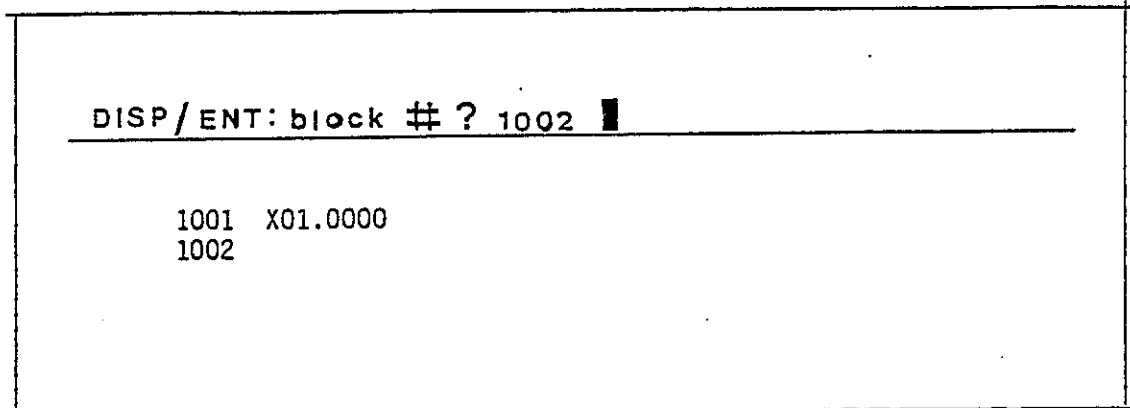


FIGURE 5.2.5 CRT DISPLAY - ENTERING PROGRAM, STEP 5

The last 4 steps are repeated for each line of the program to be stored in memory.

It is possible for the operator to display G-code and M-code definitions during this operation. If the G or M key is depressed twice in succession, a portion of those codes will be displayed in the top portion of the CRT display. The operator can continue displaying these codes by repeatedly depressing the G or M key. This can also be done in the MDI mode. In addition to the G and M listings the HELP key can be used to display aids for programming arcs, rectangular pocket clearing, rectangular and circular finish cuts, circular pocket clearing, bolt hole routines, and Z canned cycles. These aids will also be displayed in the top portion of the CRT.

No matter in what order information is entered into the control the control will reorganize the line information so that M, F, G and S instructions appear in the order they were written but prior to any R, X, Y, Z or C instructions.

5.3 PROGRAM DISPLAY

The program is displayed in much the same way as it is entered. If just displaying a program the EDIT KEY switch should be placed in the OFF position. This allows the operator to scroll through the program line by line either forward or backward. Depressing the ENT key will advance the program a line at a time for quick review. By depressing the +/- key the program is scrolled backward a line at a time. If the EDIT KEY is left in the ON position the ENT key will display the program an entry at a time. It should be noted that up to 8 lines of a program are displayed at one time on the CRT display.

EXAMPLE 5.3.1

PROGRAM DISPLAY

1 -----Program Number (can be 1-9)
0
0 ----- Line number - Causes up to 8
1 ----- lines of the program to be
----- displayed with the first
----- instruction of line 1 appearing
----- in the DISP/ENTER line on the CRT

Sample Program 5.3.1

3001 XOYO
3002 X1YO
3003 X1Y1
3004 XOY1
3005 XOYO

Key sequence for entering sample program into memory:

DISP PROG 3001 ----- This sets up program and line#
----- for entry

XO ENT YO ENT EOB Stores first line of program
X1 ENT YO ENT EOB
X1 ENT Y1 ENT EOB
XO ENT Y1 ENT EOB
XO ENT YO ENT EOB

Note: Block number will advance automatically after each
EOB is stored.

Sample program is now stored in memory.

Key sequence for displaying line 4 of sample programs. (Edit key in ON
position).

DISP PROG 3004 ----- Causes the first instruction of
----- line 4 (X) to be displayed
----- on the DISP/ENTER line with
----- lines 4 and 5 displayed in
----- their entirety just below this
----- line.

ENT ----- Causes the next instruction (Y)
----- to be displayed in DISP/ENTER
----- line and X 01.0000 will be
----- displayed immediately below
----- this line.

ENT ----- Now both entries of line 4
----- (X01.0000 and Y01.0000) are
----- displayed just below the DISP/
----- ENTER line and the cursor has
----- dropped down just behind these
----- entries.

To continue scanning the program on to Block 5, ENT can be pushed. This will advance the Block Number and start displaying Block 5.

5.3.1 DO'S AND DON'TS OF PROGRAM ENTRY

A. DO'S

1. Enter Program Number first, then 3-digit Line Number.
2. After each one to six digit entry, depress the ENT key.
3. At the end of each Block depress the EOB key. This causes the Block just entered to be stored in memory and automatically advances the Block Number.
4. An Axis can be given only one programmed move per Block.
5. When entering circular moves, the Radius, The X and Y Center(I,J) Point dimension, followed by the X and Y End Point dimensions are entered.

It is necessary that the Center Point dimensions precede the End Point dimensions in the Program Block.

6. Turn EDIT KEY to "ENABLE".
7. Enter new Block data only at END OF BLOCK.
8. $\pm$ Key when no function key has been depressed backs program up one Block.

B. DON'TS

1. Begin or end a Program or Subroutine with an arc or circular move.
2. Push EOB before all data on a given line has been entered.

5.4 PROGRAM EDITING

Before the edit features can be used, the EDIT KEY switch must be turned to the "ON" position. The EDIT KEY switch is provided as a safety feature so that no one can accidentally (or purposely) change a program.

With the EDIT KEY in the "DISABLE" position a program can be used but a program cannot be entered, changed, added to, deleted from, or erased. Therefore, to perform any of the edit features, the EDIT KEY must be in the "ENABLE" position. The edit features of the control and how they are used are as follows.

5.4.1 TO DELETE A COMMAND FROM A BLOCK

3004 XOY1

The above Block is from sample program 5.3.1. The sequence to delete Y1 from the above Block is as follows:

Key Sequence:

DISP
PROG 3 0 0 4

This will cause the X instruction in line 4 to be displayed in the DISP/ENTER line.

ENT -----

Causes the next instruction (Y) to be read up.

DEL -----

Since the DISP/ENTER line is displaying the instruction to be deleted, the DEL key will delete the Y-dimension from the Block but not change the memory.

Also the next command on the line will be displayed. In this case, there is no other information so the cursor will drop down to the line just below DISP/ENTER.

EOB -----

This will cause the edited Block to be stored back into memory. If EOB is not entered here, the memory will remain unchanged.

Block 4 now reads:

3004 XO

5.4.2 TO CHANGE A BLOCK WITHIN A PROGRAM

3005 XOYO

The above Block is from sample program 5.3.1. The sequence to change the X dimension to X0.1 is as follows.

Key Sequence:

DISP PROG	3 0 0 5	-----	This will cause the line to be displayed and the X instruction will appear on the DISP/ENTER line.
	. 1	ENT -----	This will cause the new X-dimension to be entered and the next instruc- tion will be shown on the DISP/ ENTER line.
ENT		-----	This will blank the DISP/ENTER line as there is no more informa- tion on the Block and the new line information will be displayed just below the DISP/ENTER line.
EOB		-----	This will cause the changed Block to be stored back into memory. The Block Number will advance.

5.4.3 TO DELETE AN ENTIRE BLOCK

3002 X1Y0

The above Block is from sample program 5.3.1. The sequence to delete the whole Block is as follows.

Key Sequence: -

```

DISP
PROG  3  0  0  2

DEL ----- Delete X1

DEL ----- Delete Y0

EOB ----- Store Change in Memory

```

When everything is deleted from a Block, the whole Block is deleted and all subsequent Blocks resequenced. Hence, Block 4 will now be what Block 5 was before the Delete operation.

The program will now read:

```
3001 XOY0
3002 XIY1
3003 X0
3004 X.1Y0
```

5.4.4 TO ADD INFORMATION TO A BLOCK OF A PROGRAM

```
3001 XOY0
```

The above Block is from sample program 5.3.1. The sequence to add a feedrate of 30 inches per minute to Block 001 is as follows:

- A. Display the Block to be added to
- B. Press the ENT key until all the line information is displayed just below the DISP/ENTER line and the cursor is located at the end of this information.
- C. Enter the information to be added remembering to press the ENT key after each entry.
- D. After the last entry, press the EOB key.

Key Sequence:

```
DISP
PROG 3 0 0 1 -----Program and Block Number
                        and X-displayed

ENT----- Y-Displayed

ENT----- END OF BLOCK status

F 3 0 ENT ----- Enters a Feedrate of 30 ipm
                  at end of Block 1

EOB -----Stores changed Block in Program
            and moves the feedrate to the
            start of the Block.
```

Block 001 is now::

```
3001 F30 X0 Y0
```

5.4.5 TO ADD A BLOCK WITHIN A PROGRAM

A Block will be added between Blocks 1 and 2 of Section 5.4.3.

The Key Sequence to add a Block is as follows:

- A. Press the EDIT/ADD key
- B. Enter the Program Number and Line Number to be added
- C. Press ENT and the BLOCK NUMBER will be displayed on the DISP/ENTRY line
- D. Enter the information contained in the Block being added via Program Entry (See Section V 5.2).
- E. Press the EOB key following the last entry

The new block has now been added and all subsequent Block Numbers have been incremented by one.

Key Sequence:

```
EDIT/
ADD   3   0   0   2 -----Block Number to be added

ENT -----Causes old Block 2 to be shifted
              to Block 3 and Block 3 to 4, etc.
              Block 2 will be blank and control
              is ready to accept new data.
```

```
X   1   ENT   Y   0   ENT   EOB - New Block is now entered
```

Sample program 5.3.1 now reads;

3001	F30	X0	Y0
3002	X1	Y0	
3003	X1	Y1	
3004	X0		
3005	X0.1	Y0	

5.4.6 TO ERASE A WHOLE PROGRAM

The sequence to erase the whole program given in sample program 5.3.1 is as follows:

Key Sequence:

```
CLEAR
PROG   3   0   0   1   ENT
```

This will cause the whole program 3 to be deleted from memory.

5.4.7 TO ERASE THE WHOLE MEMORY

The sequence to erase the whole memory is as follows:

Key Sequence:

```
READ  
TAPE    9    9    ENT
```

This will cause the whole memory to be cleared.

5.5 READ AND WRITE TAPE FEATURES

The control can use magnetic tape and/or paper tape readers to expand its memory storage capability and for the ease of entering programs.

The magnetic tape is typically partitioned into 6 sections, but as the control's memory is expanded, the number of sections the tape will accommodate will decrease. Each section of tape is equivalent to the entire control parts program memory capacity.

Therefore, it is not possible to overload a section of tape when writing (recording) from memory, nor is it possible to overload the memory when reading (playing) from the tape. When writing onto a section of tape, the entire memory is recorded on the section of tape selected by the operator. If the memory is not full at this time, the unused portion is recorded as blanks. Therefore, if there is anything on the section of tape selected, it will be replaced by what is in memory at the time it is recorded. This is also true of the reverse process. When reading a section of tape into memory, the memory is completely replaced by what is on the section of tape being read. If the section of tape being read is blank, the memory will be erased.

5.5.1 PROCEDURE USED TO WRITE (RECORD) A SECTION OF MAGNETIC TAPE

- A. EDIT key can be in either the "ENABLE" or "DISABLE" position
- B. All programs to be entered on the section of tape selected must be in memory. If a program is to be added to a section of tape already recorded, that section must be read into memory before the program(s) to be added are entered into memory. Once the program(s) are entered into memory the section of tape selected is ready to be written.
- C. Selector switch on the magnetic tape cassette must be switched to the "RECORD" position.
- D. Push (insert) the magnetic tape into the tape reader.

E. KEY SEQUENCE for transferring Memory to magnetic tape

WRITE
TAPE ----- Command Key

0
1 ----- Tape Sector Number (1-6)

ENT ----- Causes the transfer to be initiated

F. Display will indicate "Tape Write in Progress" and when finished it will indicate "COMPLETED", but if an E09 write tape parity occurs, there was an error during the RECORD CYCLE and the write sequence should be repeated.

G. Remove tape from reader and switch the selector on the tape "OFF" the record position as a safety precaution.

5.5.2 PROCEDURE TO READ A SECTION OF MAGNETIC TAPE INTO MEMORY

- A. EDIT key must be in the "ON" position
- B. Anything in memory that should be saved must first be recorded.
- C. Place selector switch on tape "OFF" the record position and push (insert) tape into reader.
- D. KEY SEQUENCE for reading magnetic tape into memory

READ
TAPE ----- Command

0
1 ----- Tape Sector Number (1-6)

ENT ----- Causes the transfer to be initiated

E. Display indicates "Tape Read in Progress" and when finished it will indicate "COMPLETED". But if an E08 read tape parity occurs there has been an error in the READ PROCESS and READ TAPE procedure should be repeated.

F. Remove tape from reader.

5.5.3 PROCEDURE FOR EDITING INFORMATION RECORDED ON THE MAGNETIC TAPE

- A. EDIT key must be in the "ON" position
- B. Any information of value currently in the memory should first be stored on tape.
- C. Read section of tape to be edited following the procedure in Section V 5.5.2.

- D. Edit any programs that need to be changed, erase programs or add programs as per Section V 5.4.
- E. Place selector switch on the tape to the record position and push tape back into reader.
- F. Write memory back onto tape section desired by following procedure in Section V 5.5.1.

5.5.4 DO'S AND DON'TS OF MAGNETIC TAPE READ/WRITE

A. DO'S

- 1. Always make sure that the cartridge is inserted all the way into the tape slot, or else the control will give an E11 error code.
- 2. The maximum tape READ/WRITE operation can take up to 2 minutes.

B. DON'TS

- 1. Do not attempt to pull the tape out before the control terminates a READ/WRITE operation.

5.5.5 PROCEDURE FOR READING PAPER TAPE

A. EDIT key to "ENABLE" position

- B. Make sure there is nothing of value in memory not already on tape
- C. Place paper tape in paper tape reader

D. KEY SEQUENCE for transferring paper tape into memory:

MDI ----- Mode

G

5

1

ENT ----- G-Code for Paper Tape Reader

CYCLE

START -----Initiates tape motion

Note: A "%" character on the tape or pressing CE key will terminate tape motion.

5.5.6 PAPER TAPE STANDARD
RS244-A (Original EIA Standard) or RS-358 (ASCII).

5.5.7 PAPER TAPE FORMAT

The keys that are recognized by the control are as follows.

	ASCII (Even Par.)	E.I.A.
0	00110.000	00100.000
1	10110.001	00000.001
2	10110.010	00000.010
3	00110.011	00010.011
4	10110.100	00000.100
5	00110.101	00010.101
6	00110.110	00010.110
7	10110.111	00000.111
8	10111.000	00001.000
9	00111.001	00011.001
+	00101.011	01110.000
-	00101.101	01000.000
.	00101.110	01101.011
G	01000.111	01100.111
M	01001.101	01010.100
S	01010.011	00110.010
F	11000.110	01110.110
R	11010.010	01001.001
X	11011.000	00110.111
Y	01011.001	00111.000
Z	01011.010	00101.001
%	10100.101	00001.011
N	01001.110	01000.101
E	11000.101	01110.101
(E can only appear as part of END)		
D	01000.100	01100.100
(D can only appear as part of END)		
BLANK TAPE	00000.000	00000.000
SPACE	10100.000	00010.000
CARRIAGE RETURN	10001.101	10000.000
DELETE	11111.111	11111.111
" (" COMMENT START	10101.111	00110.001
")" END OF COMMENT	10101.001	01001.010

FIGURE 5.5.1 - PAPER TAPE ASCII/EIA CODES

Any character not listed in Figure 5.5.1 is ignored by the control.

- A. The first character on the paper tape has to be a percentage (%) sign.
- B. A CARRIAGE RETURN is stored as an END OF BLOCK.
- C. Each BLOCK can have a Block Number (Format NXXX) at the beginning. The Block Numbers have to be consecutive. If they are not consecutive, the control re-numbers them in consecutive order. The first Block is always N001. If there are no Block Numbers, the control does auto-sequencing.
- D. One tape can have up to 9 programs with the ninth being reserved for subroutines. To signal the END OF PROGRAM to the control the last carriage return in the program must be followed by a three character set, "END", then a carriage return. The next program can then follow.

The maximum number of "END's" that can be on a tape is 9.

Subroutines are stored in Program 9, (after 8 "END's" have been read).

- E. The TAPE READ cycle will terminate in three conditions;
 - 1. Memory Full (ERROR #18)
 - 2. Operator presses CE key
 - 3. A percentage (%) sign character is detected on tape
- F. The TAPE READ cycle will temporarily be halted if the FEEDRATE switch is in the "HOLD" position.
- G. Tape speed is selectable at either 150 or 300 cps. See System Parameter #86 for how to select tape speeds.
- H. On a REWIND command (G50) the tape will be rewound until 16 Nulls (blanks) are detected. The control does not scan for a REWIND STOP; so if more than one "SECTORS" (set of 1 to 9 programs) are punched on one tape, they should be separated by more than 16 Nulls. (About 12" of tape is recommended.)

An M30 will also rewind tape, except that it will also generate a program stop.

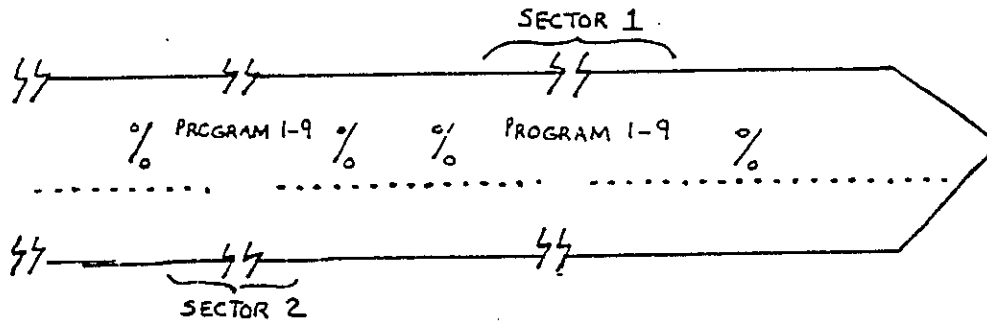


FIGURE 5.5.2 - PAPER TAPE FORMAT

EXAMPLE 5.5.1 PUNCHING A PROGRAM ON PAPER TAPE

```
% N001 F20 G1 X2 Y2 CR
N002 Y-2 CR
N003 X-2 CR
N004 Y2 CR
N005 G0 X0 Y0 CR
N006 END CR
%
```

Notice that the program was written with line numbers and spaces between commands. Neither is necessary but makes the program easier to read.

Note:

1. Each sector on the tape is De-Limited by a "%" character in Figure 5.5.2. The tape can have as many sectors as possible.
 2. If line numbers are used they must adhere to the NXXX format where XXX is the line number.
- I. Automatic selection of EIA/ASCII is accomplished by the control. This is done by comparing the codes in ASCII/EIA of the % sign which is to be the first character in a tape sector.
 - J. For tape punch operation, see RS-232 FORMAT.

5.6 RS-232 AND TTY INPUTS/OUTPUTS

- A. Whenever a G53 (EIA output) or a G54 (ASCII output) command is issued to the control, the memory contents are sent out to both the RS-232 and the TTY ports.
- B. The RS-232 port is a 3-wire interface. The three wires serve as send, receive and ground connections.
- C. The TTY port is a 4-wire interface. 2 wires serve as output, and 2 wires serve as input to the control.
- D. Whenever a G55 (input) command is issued to the control, data is scanned at both the RS-232 and the TTY ports. The user should make sure that only one port is sending to the CNC.

- E. Any time transmission or reception is to be aborted, the "CE" key can be pressed to terminate operation.
- F. BAUD RATE can be varied by system parameter #86. The following selections can be made:
- | | | |
|---------------------------------|---|--------|
| 110 baud rate system parameter | = | XXXXX0 |
| 300 baud rate system parameter | = | XXXXX1 |
| 600 baud rate system parameter | = | XXXXX2 |
| 1200 baud rate system parameter | = | XXXXX3 |

SECTION VI

TOOL OFFSETS

6.1 ENTERING TOOL OFFSETS

The control can accept offsets for up to 24 different tools. These offsets include "R" for the tool radius, and "X" and "Y" for fixture offsets, and "Z" for the tool length offsets. These offsets may be entered directly into the control or they may be entered through a program. Once entered, the tools can be accessed by tool number in either the MDI or program mode of operation. When a tool has been selected, its offsets are used to compensate any move commanded in either the MDI or program mode of operation.

6.2 ENTERING TOOL LENGTH OFFSETS

Several methods for setting tool lengths can be used. Two of the methods are going to be described in this manual.

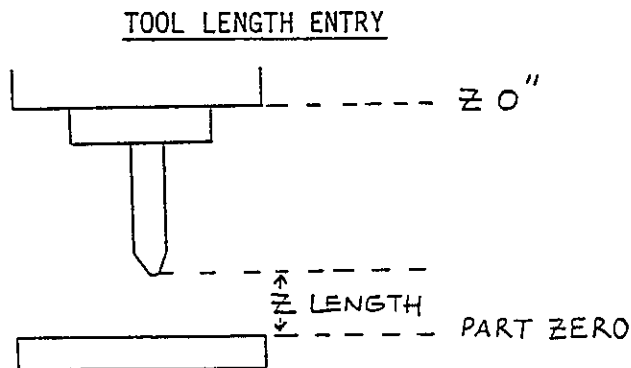


FIGURE 6.2.1 - TOOL LENGTH ENTRY

To find Z tool length in Figure 6.2.1, the tool could be loaded into the spindle (with Z at floating Zero) and using a scale, the distance from the desired part zero and the tool point could be measured. This number could then be manually loaded into the proper tool length offset by using the following procedure.

DISP
 STATUS ----- Mode
 6 ----- Page 6 is Tool Offset Table
 03 ----- Tool Offset 3
 Z ----- Z Offset or Tool Length
 -1.5 ----- Tool distance to part Zero
 (Number indicates direction
 of tool movement).

 ENT ----- Stores Z tool length in Tool
 Offset #3 Location

6.2.1 KEY SEQUENCE FOR ENTERING RADIUS, FIXTURE AND TOOL LENGTH OFFSETS

A second method to find Z tool length in Figure 6.2.1. is to load the tool into the spindle and use the following sequence:

DISP
 STATUS ----- Mode
 6 ----- Page 5 is Tool Offset Table
 04 ----- Tool Offset 4
 HDW ----- Enter Handwheel Mode
 Z ----- Z Axis Active
 Using the electronic handwheel, bring the tool down to the
 programmed zero.
 ENT ----- Stores Z Tool Length in Tool
 Offset 4
 To continue setting tools, press:
 HDW
 Z
 Position Z to load next tool. Bring next tool to programmed
 zero:
 ENT ----- Stores Z Tool Length in
 Tool Offset 5

Note: To keep knee to quill distance to a minimum, always set longest tool first. Raise (or lower) the knee to .1 to .2 clearance for this tool. Then lock knee and set all remaining tools.

6.3 TO ENTER TOOL OFFSETS FROM A PROGRAM OR FROM THE MDI MODE

1. Enter a G1XX in the program - the XX would be the tool number desired (01 - 24).
2. In the same block of the program, enter a G21 followed by the offsets for the tool selected. X and/or Y and/or Z Offsets can be entered on the same block. Radius offsets (R) have to be programmed on a block by themselves.

TOOL OFFSET ENTRY VIA THE G21 CODE USING MDI MODE

Key Sequence

```
G
1
1
0 -----Calls Tool #10
ENT
G
2
1
ENT ----- Enter Tool Offset Code
X
1
ENT ----- X 1.0 Inch
Y
1
ENT ----- Y 1.0 Inch
```

Z

-

1

.

4

ENT ----- Z -1.4 Inch

CYCLE START -----Stores X1 Y1. Z-1.4 Inch in
Tool #10

G 1 1 0 ENT CYCLE START

(This G110 is for activating the offsets. This is optional at this point, if for example, it was only desired to enter the offset into the memory and not activate them on the machine.)

Tool offsets are activated in a program or in MDI by using the G1XX code. If the tool offsets have been entered in one block of a program, they would have to be activated in a later block of the program. The G1XX code does not activate or deactivate tool radius compensation. This is activated by the G41 and G42 codes, and deactivated by the G40 code. These will be explained in a later section.

The G100 code is used to shut-off the tool offsets in either MDI or in a program. This does not effect the tool offsets themselves, as they remain in memory and can be reactivated at any time.

6.4 MODIFYING TOOL OFFSETS VIA THE MDI OR PROGRAM MODES

A G20 code is used to modify an existing tool offset.

The following example can be used to modify tool offsets in a program by simply depressing the EOB key instead of the CYCLE START pushbutton.

NOTE: When entering or modifying tool offsets in a program it must be remembered that tool radius offsets must not appear on the same program line as X, Y and Z offsets.

Key Sequence

G

1

1

0 -----Calls Tool #10

ENT

G

2

0

ENT ----- Modify Tool Offset Code

X

.

5

ENT ----- X .5 Inch

Y

-

.

5

ENT ----- Y -.5 Inch

Z

.

0

1

5

ENT -----Z .015 Inch

CYCLE START ----- Modifies Tool Offset #10 by
adding .5" to the X offset of
1", subtracting .5" from the
Y offset of 1" and adding .015"
to the Z offset of -1.4".

To activate modified tool offset, use key sequence:

G 1 1 0 ENT CYCLE START

SECTION VII

RUNNING BASIC PROGRAMS

7.1 RUNNING BASIC PROGRAMS

To RUN a program in AUTO mode, use the following procedure:

- A. Press RUN
- B. Press NXXX Where N represents Program Number
and XXX represents Line Number
- C. Press CYCLE START

EXAMPLE 7.1.1

Key Sequence to run program #3:

```

RUN      3      0      0      1  CYCLE
                                START

```

The machine will RUN Program 3 starting at Block Number 1.

The machine will continue until it reaches an M06, M00 or M02.

Note: If AUTO/BLOCK BY BLOCK selector switch is in "BLOCK BY BLOCK" then at the end of the block of data, the control will wait for another CYCLE START to continue.

7.1.1 DRY RUN

To DRY RUN a program, follow the procedure in Section 7.1.

The difference between DRY RUN and RUN is that in DRY RUN a minimum feedrate is forced by the control. This means that the program may not be run at less than 75 IPM unless the FEEDRATE OVERRIDE switch is turned to less than 100%. The purpose of DRY RUN is to test a program at a fast feedrate. This eliminates wasted time waiting for a long move at a slow feedrate.

Note: M07 and M08 are ignored in DRY RUN.

7.2 CODES WHICH ARE ACTIVATED WHEN THE CONTROL IS FIRST TURNED "ON"

1. G01 - The control is automatically programmed for a feedrate of 20IPM when it is first turned "ON", which means any programmed move will occur at 20IPM unless a new feedrate is entered or a G00 code is entered for a rapid move.

2. G03 - CCW Circle
3. G10 - Display Z-axis
4. G17 - XY Plane is active for Circular moves
5. G30 - Continuous Path Mode (corner rounding)
6. G40 - Tool Radius Compensation is "OFF" when control is first turned "ON"
7. G60 - The mirror images are "OFF"
8. G70 - All dimensions are entered in inches
9. G90 - Control assumes absolute programming unless it is programmed for incremental
10. Floating Zero - is at GRID ZERO after the GRID ZERO IS initially set
11. G101 - The Tool #1 Offsets are active upon turn "ON"

The state of these codes upon turn "ON" explain and dictate many of the initial set-up procedures necessary for running a program.

The operator must know the status of these codes before running a program, or take them into consideration when writing a program.

7.3 BASIC SET-UP PRIOR TO RUNNING A PROGRAM

This basic set-up procedure should be performed before a program can be safely run. The control must be in the absolute mode of operation (G90) to set tool offsets and floating zeroes. Absolute mode can be verified by checking DISPLAY STATUS Page 0.

1. Make sure the X, Y and Z Floating Zeroes are set properly
2. Set the appropriate Tool Offsets and activate the corresponding tool (See Section XI) if this is not done in the program.
3. Set the R-Plane at an appropriate value, if this is not done in the program. This is done with the G300 code followed by a PXX.XXXX when XX.XXXX is the distance of the R-Plane above the workpiece. This is typically a distance which will clear any fixture or workpiece obstructions. The R-Plane is used as a tool retract location (G99) for clearing the tool of the work area prior to making rapid moves. Whenever a new tool, with a different length offset, is established with the G1XX code the R-Plane must be re-established. Otherwise, it remains at its original position and if the new tool is longer the G99 code could actually drive the tool into the workpiece.

4. The G70 (Inch) and G71 (Metric) codes cannot be programmed and should be selected via the MDI mode prior to running a program.

A program should first be run in DRY RUN with the AUTO/BLOCK BY BLOCK switch in "BLOCK BY BLOCK" position with the tool removed from the spindle; go through the program one block at a time checking all axes for proper moves and feedrates. Also check for proper G and M code operations.

7.4 POINT TO POINT PROGRAMMING

For the following programming examples we will assume that the Centurion control does not have any of the features not already discussed. Features like autoroutines, canned cycles, tool radius compensation, etc. will be discussed in later sections. Point to Point programming includes hole patterns and rectangular milling.

The following program is a simple matrix of holes identified by the letters A through I.

EXAMPLE 7.4.1

TOOL #: 1 (1/2" drill)

PART #: Random Hole Drilling Pattern

POS.	BLOCK #	LINE ENTRY INFO:
A	1001	G0, G99, X 0.5, Y2.0
	1002	F6, G1, M3, M7, Z-.25
B	1003	G0, G99, Y4.5
	1004	G1, Z-.25
C	1005	G0, G99, X1.0, Y1.0
	1006	G1, Z-.25
D	1007	G0, G99, X1.5, Y4.0
	1008	G1, Z-.25
E	1009	G0, G99, X2.0, Y.5
	1010	G1, Z-.25
F	1011	G0, G99, X3.0, Y1.5
	1012	G1, Z-.25
G	1013	G0, G99, Y3.0
	1014	G1, Z-.25
H	1015	G0, G99, X3.5, Y4.5
	1016	G1, Z-.25
I	1017	G0, G99, X4.5, Y2.0
	1018	G1, Z-.25
	1019	G0, G99, M5, M9, X0, Y0

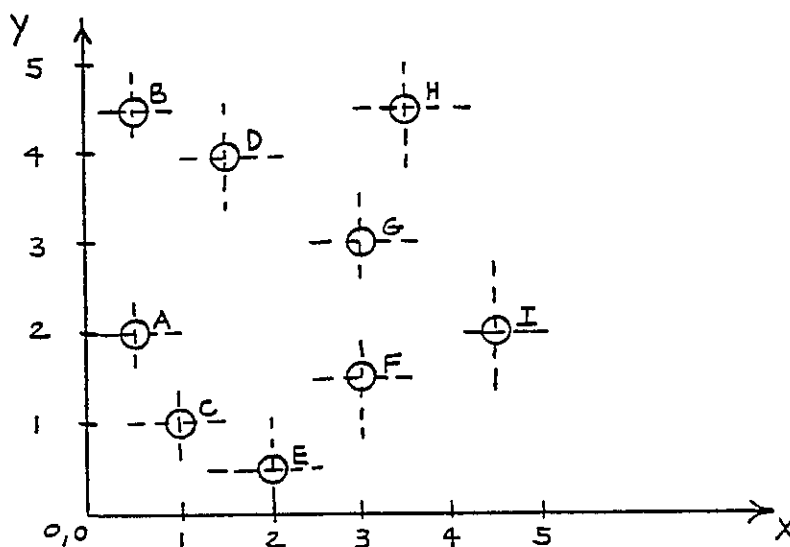


FIGURE 7.4.1

In setting up this program the operator must do the following:

1. First, enter the program into the control memory as described in Section V 5.2.
2. Next, locate and set the Floating Zero at the corner of the piece as indicated by 0 on the print. Reference Section IV 4.2.
3. Set Z tool offset dimension for the drill being used. Reference Section VI 6.2.

4. Activate the tool # being used by depressing KEY SEQUENCE

MDI G 1 0 1 ENT CYCLE START

5. Establish an R-Plane .100 inch above the workpiece using a G300
P.1 KEY SEQUENCE

MDI G300 ENT P.1 ENT CYCLE START

6. The program can now be checked by removing the tool, depressing the DRY RUN key, and selecting program number 1 and line number 001 followed by the CYCLE START button. This will cause the spindle to turn on, but not the coolant, and the machine will rapid to each position where the Z-axis will feed down at 75 ipm followed by a rapid move back to the R-Plane. At the end of the program the spindle will shut-off, Z will return to the R-Plane and X and Y will rapid to Floating Zero.

7. If the program checks out OK, the tool can be reinserted and the program can be run by doing the following:

KEY SEQUENCE

RUN 1 0 0 1 CYCLE START

This causes the program to run with the odd numbered lines from 1 to 17 causing the machine to locate the tool at each hole position and the even numbered lines from 2 to 18 causing the respective holes to be drilled.

The M3 in line 2 turns the spindle on and M7 turns the spray mist coolant on. In line 19 the G0 forces a rapid feedrate, the G99 brings the Z-axis up to the R-Plane, the M5 code shuts the spindle off, the M9 shuts the coolant off, and the X0 and Y0 returns the X and Y axes to their Floating Zero position.

7.5 LINEAR INTERPOLATION OR ANGULAR MILLING

This program is basically the same as the hole drilling pattern.

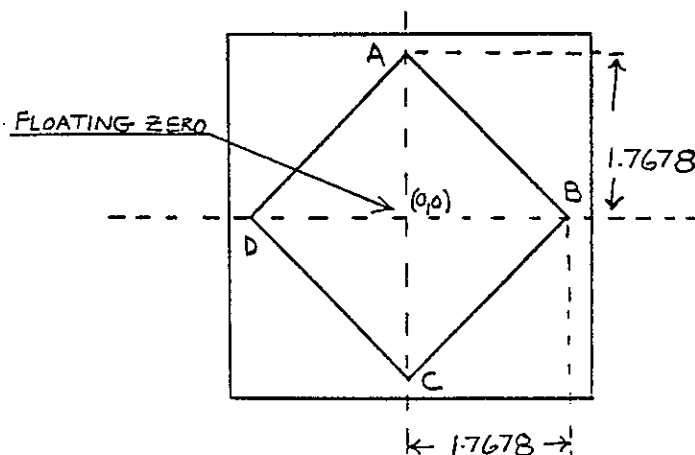


FIGURE 7.5.1 -

EXAMPLE 7.5.1

TOOL #: 2 ($\frac{1}{2}$ " mill)

PART #: Square/Diamond Pattern
Tape Section: 2

POS.	BLOCK #	LINE ENTRY INFO.
A	2001	G0, G99, X0, Y1.7678
A	2002	F10, G1, M3, M8, Z-.125
B	2003	X1.7678, Y0, Z-.250
C	2004	X0, Y-1.7678, Z-.125
D	2005	X-1.7678, Y0, Z-.250
A	2006	X0, Y1.7678, Z-.125
O	2007	G0, G98, M5, M9

The same set-up procedure applies for this example as for example 7.4.1

1. Enter program. (See Section V 5.2)
2. Set Floating Zero. (See Section IV 4.2)
3. Actuate Tool #02 and set Z (tool length) Offset (See Section VI 6.2)
4. Set R-Plane. (See Section VII 7.3)
5. Dry Run Program with Tool Removed (as in program #1).
6. Run Program

KEY SEQUENCE

RUN 2 0 0 1 CYCLE START

If the program were run at a higher feedrate, say above 30 ipm, it may be necessary to use the G31 code to retain accuracy at the corners. This program will run from point A to B to C to D and the G98 in line 7 will cause all axes to return to their Floating Zero position, at the rapid feedrate, with Z-axis positioning first.

7.6 CIRCULAR INTERPOLATION PROGRAMMING

The Centurion will accomplish circles and arcs at a feedrate up to 100 ipm. If an arc is programmed above 100 ipm it will automatically be performed at 100 ipm. The last programmed move in any program(s) must be a linear move. It can be as small as .0001". This enables the cutter compensation routine to calculate displaced cutter path correctly.

When programming an arc or circle it is necessary to give the direction, either clockwise or counterclockwise (G2 or G3), radius, center point dimensions, and end point dimensions, in that order. (See Example 7.6.1). If the direction was given in an earlier line and is the same, it is unnecessary to repeat the command. It is also necessary to position the tool to the start point of the arc before the arc command is given.

CIRCLE PROGRAM

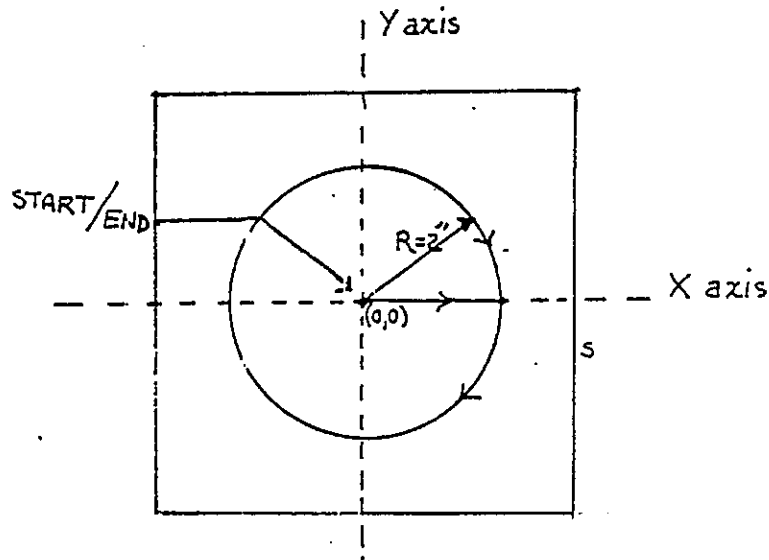


FIGURE 7.6.1

EXAMPLE 7.6.1

TOOL #: 3 (3/8" mill)

PART #: Circle Pattern

POS.	BLOCK #	LINE ENTRY INFO.
A	3001	G103, G300, P.3
A	3002	G0, G99, X0, Y0
B	3003	F10, G1, M3, M7, X1.8125, Y0, Z-.125
B	3004	G2, R1.8125, X0, Y0, X1.8125, Y0
A	3005	G0, M5, M9, G99, X0, Y0

Set up for this program is accomplished in line 3001. This program begins at the center, moves the tool out to the start point on the circle in line 3003, does the circle in line 3004, and rapids back to the center of the circle with Z-axis returning to the R-Plane in line 3005.

Note: At the end of a program or upon an M02 command the control will automatically shut-off the spindle, coolant, cutter compensation, mirror image or axis interchange, and any active canned cycle. However, it is good practice to return the control to its normal static state at the end of each program. It should be noted that the control will not change the active plane once established. Again it would be good practice to return the control to the XY plane (G17) of operation when finished programming in one of the other planes.

NOTES:

1. The first and last moves of a program must be linear.
2. Feedrates are not active unless followed by a G1.
3. All G3XX and G4XX codes must be immediately followed by an appropriate parameter (P) value..
4. G20 and G21 codes must be immediately followed by tool offset values; however the radius (R) offset cannot appear on the same program line with X,Y, or Z offsets.
5. A G92 code must be immediately followed by the floating zero position desired as referenced to current tool position.
6. Do not follow a G8X4 code with an M code as this will cause the subroutine to be looped on.
7. Make sure that unused parameters are zeroed (especially parameters 3,4,9,23 and 24). When these are used it would be good practice to zero them at the end of the program.
8. Only one G3XX or G4XX code can appear on a single program line and cannot appear on the same line with an R,X,Y,Z or C instruction.
9. Include as much of the program set-up information in the first few lines of the program as possible.
Such information as:
 - a) feedrate and spindle speed
 - b) select feed (G1) or rapid (G0) mode
 - c) select the appropriate plane (G13-19)
 - d) call up the appropriate tool offsets (G1XX)
 - e) establish the R-plane
 - f) establish autoroutines and canned cycle parameters

SECTION VIII

USE OF CANNED CYCLES

THE FOLLOWING CANNED CYCLES ARE PERFORMED BY THE CENTURION CONTROL.

- G79 - Re-enable Canned Cycle
- G80 - Deactivate Canned Cycle
- G81 - Drill Cycle
- G82 - Drill with Dwell Cycle
- G83 - Peck and Drill Cycle I
- G84 - Tapping Cycle
- G85 - Bore Cycle
- G86 - Milling Cycle
- G87 - Peck and Drill Cycle II (does not retract to B-Plane after each increment)
- G89 - Bore with Dwell Cycle

For detailed explanations of each canned cycle reference Section II 2.6.

8.1 TURNING CANNED CYCLES ON AND OFF

Important notes concerning the canned cycles.

1. Once entered a canned cycle remains active until it is shut off with a G80 code or another canned cycle is entered.
2. Only one canned cycle can be active at a time.
3. When a canned cycle is deactivated by the G80 code it remains in storage so that it can be re-enabled by the G79 code.
4. A canned cycle is active upon entry and need not be enabled by the G79 code unless it was previously turned off by the G80 code.
5. The Z depths for canned cycles can be specified with a G81-89 code and P dimension or with a G308 code and a P dimension. This Z value can be added to or subtracted from by using the G408 code. (See Section VI 6.4).
6. Do not establish a canned cycle while in the incremental mode G91. You must be in the absolute mode (G90) to do this. Once a canned cycle is established it can be turned on or off in any mode of operation.
7. A Radius (R) dimension cannot appear on the same program line with any of the G80-G89 command cycle commands.
8. Parameter #24 (G324, P.XXXX) is used with the canned cycles to give these cycles their own special feedrate regardless of the program feedrate. If Parameter #24 is zero the canned cycles will run at the program feedrate. G324, P.0155 is a feedrate of 15.5 ipm.

9. Parameter #23 (G323, PXXX.X) is used with canned cycles G82, G84, and G89 to establish the dwell time (max. dwell is 999.9 seconds).
10. Parameters 6 (G306) and 8 (G308) must have signed values with respect to the Z level at which the tool length is set (i.e. if the tool length is set by touching off at the top of the work-piece; parameters 6 and 8 will have negative values, but if the tool length is set by touching-off the table surface these values will be positive). Parameter 7 (G307) is unsigned meaning it will always have a positive value.

8.2 EXAMPLES SHOWING THE USE OF SOME CANNED CYCLES

EXAMPLE 8.2.1

Example 7.4.1 will now be re-written using canned drilling and tapping routines.

<u>POS.</u>	<u>BLOCK #</u>	<u>LINE ENTRY INFO.</u>
	1001	F6, G0, G99, G101, S1400, M3, M7, G300, P.1 Calls Tool #, sets R-Plane and starts spindle and coolant
	1002	G81, P-0.25 Enables drill cycle and sets drill depth
A	1003	X.5, Y2
B	1004	Y4.5
C	1005	X1, Y1
D	1006	X1.5, Y4
E	1007	X2, Y0.5
F	1008	X3, Y1.5
G	1009	Y3
H	1010	X3.5, Y4
I	1011	X4.5, Y2
	1012	M5, M9, G80, G98
		Return all axes to Floating Zero
		Deactivate Canned Cycle

Notes about EXAMPLE 8.2.1

Line 1001 - The feedrate command (F6) must precede the G0 command. If this format is violated all moves will be at 6 ipm.

Line 1002 - Depending on the desired operation, any canned cycle G81 through G85 could be used.

Line 1003 - Positions of the holes
1011

Line 1013 - G80 deactivates the canned cycle. If this was not done Z would feed down after the G98 return to home command.

EXAMPLE 8.2.2

Example 7.5.1 will be revised to mill the diamond to a depth of .250 inch using a G86 mill canned cycle.

<u>POS.</u>	<u>BLOCK #</u>	<u>LINE ENTRY INFO.</u>
	2001	F10, G0, G99, S2000, M3, M7, G300, P.1
	2002	G86, Z-.250 Enables Mill Cycle
A	2003	X0, Y1.7678)
B	2004	G1, X1.7678, Y0)
C	2005	X0, Y-1.7678) Dimensional Data
D	2006	X-1.7678, Y0)
A	2007	X0, Y1.7678
	2008	G80, G99, M5, M9, G98
		Canned Cycle rapid to R-Plane rapid to X0Y0Z0

Notes about EXAMPLE 8.2.2

When the program is run Z will feed to a depth of .250 inch after the X and Y move of line 2003. Z-axis will remain at this depth until the cycle is cancelled with the G80 and returned to the R-Plane with a G99.

SECTION IX

AUTOROUTINES

The autoroutines are a few selected basic patterns which are used very often on milling machines. These basic patterns are the circular and rectangular finish cuts, circular and rectangular pocket clearing, and bolt hole routines. Used with the flexibility of the parameters, the autoroutines have a wide variety of applications which will greatly simplify many programming tasks.

CAUTION: If control is EMERGENCY STOPPED in an auto routine, the program must be restarted at the Block Number containing the parameters necessary to run the autoroutine.

In the following examples, the tool radius is assumed to be zero.

9.1 G26 - CIRCULAR FINISH

The G2, G3, G41 and G42 codes are used together to determine not only the direction of cut, but whether the cut is to be an outside or inside cut. Parameters are used which will determine the dimensions of the circle. For a circle in the XY plane the X and Y will be twice the R dimension. For outside finish cuts, the radius of the tool will automatically be figured by the control to determine the proper starting point.

If a tool radius is specified, cutter compensation can be used in all autoroutines. The control will automatically decrease or increase the tool path by the diameter of the tool.

ALL AUTOROUTINES MUST BE ACTIVATED WITH THE TOOL AT THE CENTER OF THE ROUTINE.

The G86 milling cycle is used to specify cutter depth and allow automatic down feed of the Z-axis at the proper point in the autoroutine.

Note: For additional information on loading parameters, See Section II 2.6.

Figures 9.1.1 to 9.1.4 will show the tool path of the following program with the only change being made in Line 005.

BLOCK # BLOCK ENTRY INFO

```

1001  G104, M06, G306, P-.5
1002  G300, P.1
1003  G307, P.5
1004  G0, X0, Y0
1005  SEE SPECIFIC EXAMPLE
1006  G301, P2
1007  G302, P2
1008  G303, P1
1009  S2000, M3
1010  G86, P -1.0
1011  G26, G80, G40, G99, G100
1012  G98

```

```

Line 001      Activates Tool #4, and tool change and sets the
               first Z increment (G306) at-.5"
Line 002      Set R-Plane
Line 003      Set Z increment depth at .5"
Line 004      Rapids cutter to center of pocket
Line 005      SEE SPECIFIC EXAMPLE
Line 006      Enters X pocket dimension (2 inches)
Line 007      Enters Y pocket dimension (2 inches)
Line 008      Enters R (radius) dimension of the circle (1 inch)
Line 009      Start spindle
Line 010      Enable mill canned cycle and sets mill depth at
               -1 inch
Line 011      G26 will cause the X and Y axes to describe the
               pattern shown. Cancel tool offset. Rapid Z to
               R-Plane
Line 012      Rapids XY&Z to Floating Zero

```

Note: The G306 and G307 parameters in lines 1 and 3 together with the depth of cut parameter #8 work with the milling cycle (G86) in all auto-routines to determine how many passes will be made before the auto-routine reaches the programmed depth. In this example, the first pass will be made at a depth of .5 inch (G306) and the second pass will be at the finish depth since parameter #7 (G307 is also .5 inch and the final depth is 1 inch which was established with the G86 code in line 10.

*1005 F10, G2, G42 (Enters Feedrate of 10ipm, selects CW circle and right cutter compensation)

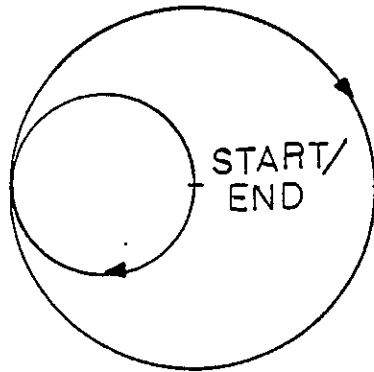


FIGURE 9.1.1 INSIDE CLOCKWISE FINISH CIRCLE

*1005 F10, G3, G41 (Enters Feedrate of 10ipm, CCW direction, and left cutter compensation)

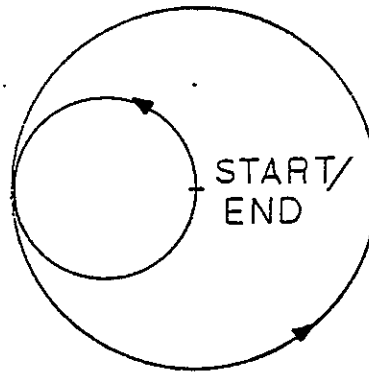


FIGURE 9.1.2 INSIDE COUNTERCLOCKWISE FINISH CIRCLE

*1005 F10, G2, G41 (Enters Feedrate, CW direction, and left cutter compensation)

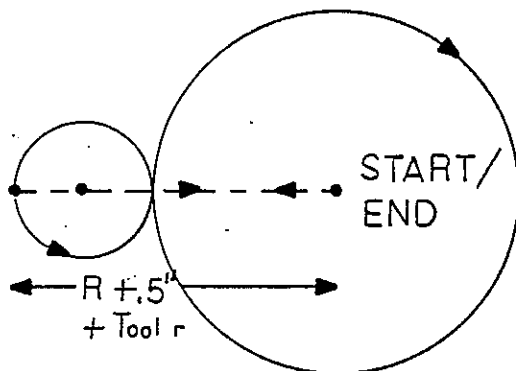


FIGURE 9.1.3 OUTSIDE CLOCKWISE FINISH CIRCLE

*1005 F10, G3, G42 (Enters Feedrate CCW direction, and right cutter compensation)

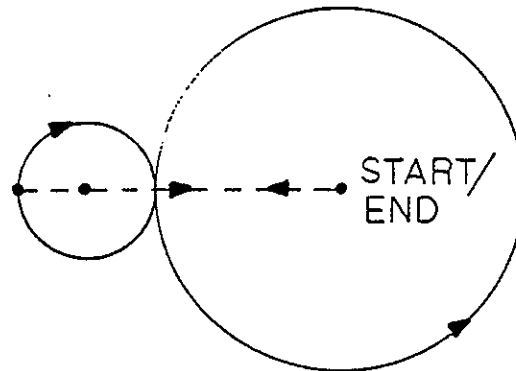


FIGURE 9.1.4 OUTSIDE COUNTERCLOCKWISE FINISH CIRCLE

9.2 G26 - RECTANGULAR FINISH WITH CIRCULAR OR SQUARE CORNERS

The rectangular finish with circular corners is accomplished by the same routine as the circular finish routine. The difference is that the programmed radius refers to the radius of the corners, while the X (G301) and Y (G302) dimensions are equal to the total length and width dimension of the rectangle.

Therefore, for the G26 code to define a rectangle, the X (G301) and Y (G302) dimensions must be greater than the radius dimensions. If the radius of the corners is set at zero, an outside pocket will have square corners and an inside pocket will have round corners equal to the cutter radius. All examples will be shown with circular corners.

EXAMPLE 9.2.1 to 9.2.4
X side less or equal to Y side

```

2001 G105, M06, G306, P -.5
2002 G300, P1
2003 G307, P.5
2004 G0, X0, Y0
2005 *SEE SPECIFIC EXAMPLE
2006 G301, P2
2007 G302, P3
2008 G303, P.5
2009 S1100, M3
2010 G86, Z-1.0
2011 G26, G80, G40, G99, G100
2012 G98

```

EXAMPLE 9.2.5 to 9.2.8
X side greater than Y side

```

3001 G105, M06, G306, P -.5
3002 G300, P1
3003 G307, P.5
3004 G0, X0, Y0
3005 *SEE SPECIFIC EXAMPLE
3006 G301, P2
3007 G302, P3
3008 G303, P.5
3009 S1100, M3
3010 G86, Z-1.0
3011 G26, G80, G40, G99, G100
3012 G98

```

```

Line 001      Activates Tool #5, tool change, sets the first
               depth parameter (G306) at-.5 inch
Line 002      Sets R-Plane
Line 003      Sets Z increment parameter
Line 004      Rapids tool to center of pocket
Line 005      *SEE SPECIFIC EXAMPLE
Line 006      Enters X pocket dimension
Line 007      Enters Y pocket dimension
Line 008      Enters R corner dimension
Line 009      Starts spindle
Line 010      Enable mill cycle and sets final depth at 1.0 inch
Line 011      G26 performs the autoroutine. The tool is brought
               down at the start point and parameters 1, 2 and 3
               describe the above shapes. Cancel tool offset.
Line 012      Rapid to Floating Zero

```

Note: The Z increment parameters in lines 1 and 3 together with the final depth in line 10 determine the number of passes it will take to complete each autoroutine. In this case it will take 2 passes to reach the final depth of 1 inch.

BLOCK #	LINE ENTRY INFO.
---------	------------------

*2005	F10, G2, G42 Enters feedrate of 10 IPM, CW circle and <u>right</u> cutter compensation
-------	---

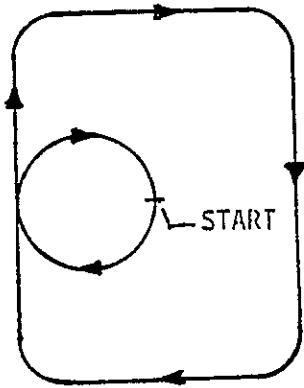


FIGURE 9.2.1
INSIDE, CLOCKWISE, RECTANGULAR
X SIDE < Y SIDE

BLOCK #	LINE ENTRY INFO.
---------	------------------

*3005	F10, G2, G42 Enters feedrate of 10 IPM, CW direction and <u>right</u> cutter compensation
-------	--

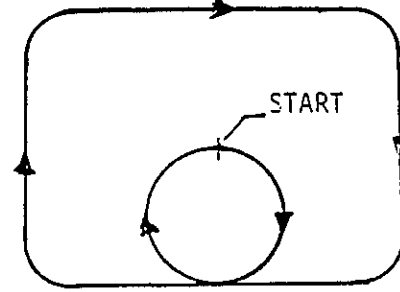


FIGURE 9.2.5
INSIDE, CLOCKWISE, RECTANGULAR
X SIDE > Y SIDE

EXAMPLE

BLOCK #	LINE ENTRY INFO.
---------	------------------

#2005	F10, G3, G41 Enters feedrate of 10 IPM, CCW direction, <u>left</u> cutter compensation
-------	---

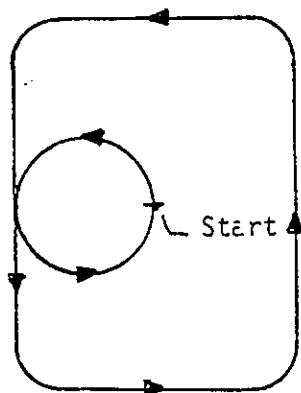


FIGURE 9.2.2
INSIDE, COUNTERCLOCKWISE,
RECTANGULAR FINISH WITH
CIRCULAR CORNERS AND THE
X SIDE < Y SIDE

BLOCK #	LINE ENTRY INFO.
---------	------------------

*3005	F10, G3, G41 Enters feedrate of 10 IPM, CCW direction and <u>left</u> cutter compensation
-------	--

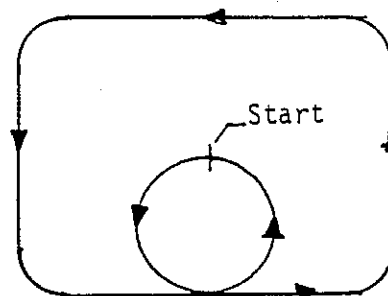


FIGURE 9.2.6
INSIDE, COUNTERCLOCKWISE,
RECTANGULAR FINISH WITH
CIRCULAR CORNERS AND THE
X SIDE > Y SIDE

EXAMPLE

BLOCK #	LINE ENTRY INFO.
*2005	F10, G2, G41 Enters feedrate of 10 IPM, CW direction and <u>left</u> cutter compensation

BLOCK #	LINE ENTRY INFO.
*3005	F10, G2, G41 Enters feedrate of 10 IPM, CW direction, and <u>left</u> cutter compensation

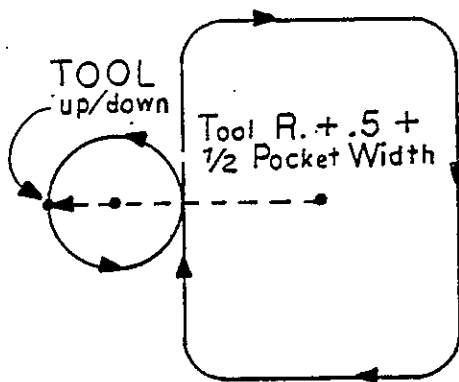


FIGURE 9.2.3
OUTSIDE, CLOCKWISE,
RECTANGULAR FINISH
WITH CIRCULAR CORNERS
X SIDE < Y SIDE

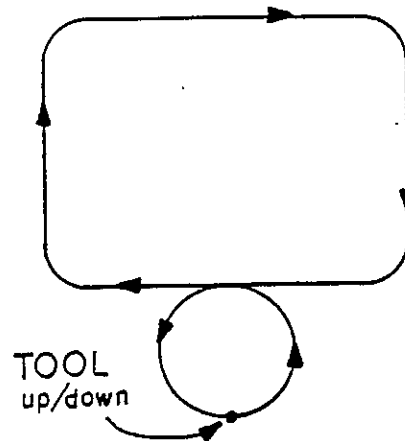


FIGURE 9.2.7
OUTSIDE, CLOCKWISE,
RECTANGULAR FINISH
WITH CIRCULAR CORNERS
X SIDE > Y SIDE

EXAMPLE

BLOCK #	LINE ENTRY INFO.
*2005	F10, G3, G42 Enters feedrate of 10 IPM, CCW direction and <u>right</u> cutter compensation

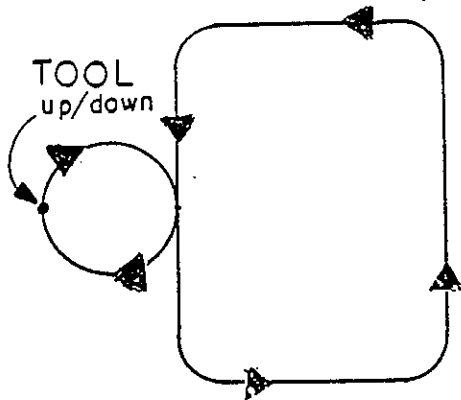


FIGURE 9.2.4
OUTSIDE, COUNTERCLOCKWISE, REC-
TANGULAR FINISH WITH CIRCULAR
CORNERS
X SIDE < Y SIDE

BLOCK #	LINE ENTRY INFO.
*3005	F10, G3, G42 Enters feedrate of 10 IPM, CCW direction and <u>right</u> cutter compensation

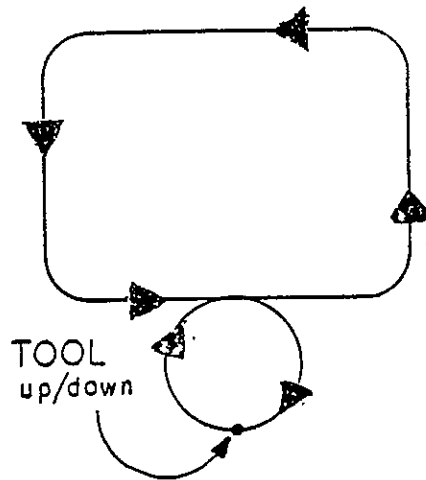


FIGURE 9.2.8
OUTSIDE, COUNTERCLOCKWISE, RECTANGULAR
FINISH WITH CIRCULAR FINISH
X SIDE > Y SIDE

9.3 G27 - RECTANGULAR POCKET CLEARING

This routine requires the pocket dimensions to be entered via a command in exactly the same format as the G26 routine. In addition, a Y increment needs to be entered via a G305 command. This increment will step the tool across the part by the value entered.

EXAMPLE

RECTANGULAR POCKET CLEARING

BLOCK #	BLOCK ENTRY INFO
3001	G106, M06
3002	G300, P.1
3003	G306, P-.15
3004	G307, P.1
3005	G308, P-.58
3006	G0, X0, Y0
3007	F5, G42
3008	G301, P3
3009	G302, P2
3010	G304, P.01
3011	G305, P.25
3012	S1800, M3, M7
3013	G86
3014	G27, G80, G40, G99, G100
3015	G98

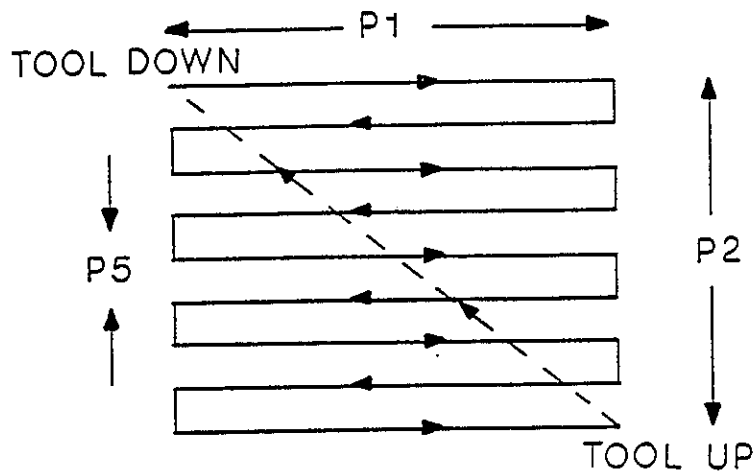


FIGURE 9.3.1 RECTANGULAR POCKET CLEARING

Line 001	Activates tool #6, tool change
Line 002	Sets R-Plane
Line 003	Sets depth of first pass
Line 004	Z-increment to depth for each pass
Line 005	Z-depth
Line 006	Rapids tool to center of pocket
Line 007	Enters feedrate of 5 IPM
Line 008	Enters X Pocket dimensions (3 inches)
Line 009	Enters Y pocket dimensions (2 inches)
Line 010	Enters dimension of finish stock (.01 inch)
Line 011	Enters step over distance for each pass (.25 inch)
Line 012	Starts spindle
Line 013	Enable Mill cycle
Line 014	G27 initiates the pocket clearing routine. The X and Y axes will rapid to the upper left corner of the pocket where Z will feed down to the depth specified by parameter #6 (G306). The X and Y axes will now describe the pocket as shown in figure 9.3.1. After the final pass across the pocket, Z rapids to the R-plane before X and Y rapid back to the start point. This procedure is now repeated to the next depth, specified by parameter #7 (G307), until the final depth, as specified by parameter #8 (G308) is reached. Since the finish stock was set at .01 inch (G304) the cutter will stay away from the pocket walls by this amount.
Line 015	Rapids to Floating Zero

EXAMPLE 9.3.2RECTANGULAR INSIDE POCKET CLEARING WITH A FINISH PASS

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
1001	G106, M06
1002	G300, P.1
1003	G306, P-.15
1004	G307, P.1
1005	G308, P-.58
1006	G0, X0, Y0
1007	F5, G42, G2
1008	G301, P2
1009	G302, P3
1010	G303, P0
1011	G304, P.01
1012	G305, P.25
1013	S1600, M3, M7
1014	G86
1015	G27, G26, G80, G40, G99, G100
1016	G98

Example 9.3.2 will work exactly as Example 9.3.1 . However, after the G27 cycle is completed, a G26 cycle will be initiated using the same pocket dimensions that G27 used. The G26 routine will clean-up the .01 inch, left by the G27 routine automatically.

Note: In each of the last 2 examples it will take 6 passes for the autoroutine to reach their finish depth.

Line 001	Activates Tool #6, tool change
Line 002	Sets R-Plane
Line 003	Sets depth of 1st pass
Line 004	Z-increment to depth for each pass
Line 005	Z-depth
Line 006	Rapids tool to center of pocket
Line 007	Enters feedrate of 5 IPM. It also instructs the G26 cycle that the pocket is an inside pocket (G42,G2)
Line 008	Enters X pocket dimension (2 inches)
Line 009	Enters Y pocket dimension (3 inches)
Line 010	Enters R corner dimension (0 inches)
Line 011	Enters amount of finish stock (.01 inch)
Line 012	Enters increment for each pass (.25 inch)
Line 013	Start spindle
Line 014	Enable Mill cycle
Line 015	Instructs control to perform a G27 cycle followed by a G26 cycle. Deactivate tool offset.

9.4 G28 - CIRCULAR POCKET CLEARING WITH FINISH CUT

For this routine, only the radius dimension need be given with the G303 code. The X increment given with the G305 code will determine the width of cut to be taken with each circular pass.

EXAMPLE 9.4.1

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
4001	G107, M06
4002	G300, P.05
4003	G306, P-.4
4004	G307 P-.15
4005	G308, P-.62
4006	G0, X0, Y0
4007	*SEE SPECIFIC EXAMPLE
4008	G301, P2
4009	G302, P2
4010	G303, P1
4011	G304, P.01
4012	G305, P.2
4013	S1400, M3, M7
4014	G86
4015	G28, G26, G80, G40, G99, G100
4016	M5, M9, G98

The G28 and G26 pass will look like Figure 9.4.1 and 9.4.2

Line 001	Activates Tool #7, tool change
Line 002	Sets R-Plane
Line 003	Z-depth for 1st pass
Line 004	Z-increment to depth for each pass
Line 005	Z-depth
Line 006	Rapid tool to center of pocket
Line 007	Enters feedrate of 5 IPM, G2 sets clockwise G42 right cutter compensation ON
Line 008	Enters X pocket dimension (2 inches)
Line 009	Enters Y pocket dimension (2 inches)
Line 010	G303 sets radius of the pocket (1 inch)
Line 011	G304 enters finish stock to the left (.01 inch)
Line 012	G305 sets cutter out feed distance (.2 inch)
Line 013	Start spindle
Line 014	Enable Mill cycle
Line 015	G28 will cause the X and Y axes to spiral out to the programmed diameter minus .01 inch (G304), then G26 takes one finish pass at programmed diameter and returns to the start point. Cancel tool offset

Note: Since parameter #6 is .4 inch and parameter #7 is .15 inch it will take these autoroutines 3 passes to reach the final depth of .62 inch.

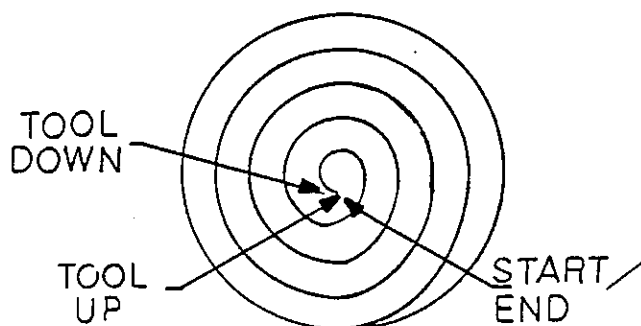


FIGURE 9.4.1
CW CIRCULAR POCKET
CLEARING

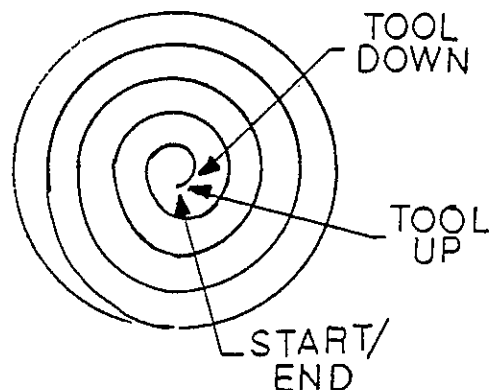


FIGURE 9.4.2
CCW CIRCULAR POCKET
CLEARING

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
*4007	F10, G2, G42
Block 007	Enters feedrate of 10 IPM, CW circle, and turns ON <u>right</u> cutter compensation

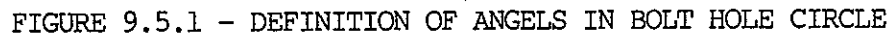
<u>BLOCK #</u>	<u>BLOCK ENTRY INFO.</u>
*4007	F10, G3, G41
Block 007	Enters feedrate of 10 IPM, CCW Circle and turns ON <u>left</u> cutter compensation

9.5 BOLT HOLE CIRCLE ROUTINE

The bolt hole circle routine needs to have the number of holes in the pattern, the radius of the pattern, and the starting angle ($\emptyset$) of the first hole. It will automatically calculate the position of each hole and position to these holes in straight moves.

The number of holes in the pattern is given with the G9XX code (XX = number of holes), the radius of the pattern with the G303 code, and the starting angle ($\emptyset$) of the pattern with the G310 code followed by a p entry equal to $\emptyset/10$ where the $\emptyset$ is equal to the starting angle in degrees. The starting angle is determined by the angle from the 3 o'clock position of the circle moving counterclockwise to the first hole location.

The 12 o'clock position is at 90° , the 9 o'clock position at 180° , the 6 o'clock position at 270° , and the 3 o'clock position is 360° or 0° (See Figure 9.5.1). If the starting angle is not entered, the last G310, P entry will be used for the starting angle.



<u>BLOCK #</u>	<u>BLOCK ENTRY INFO.</u>
1001	G108, M6
1002	G300, P.1
1003	G308, P-.75
1004	G3, G0, X0, Y0
1005	G303, P2
1006	G310, P31.5
1007	G311 P5.00
1008	G81
1009	G905, G80, G99, G100
1010	M5, M9, G98

01.07

If a G2 were used in Block 001 of this Example, Hole #5 would become the 2nd hole drilled and Hole #2 would be the last hole drilled.

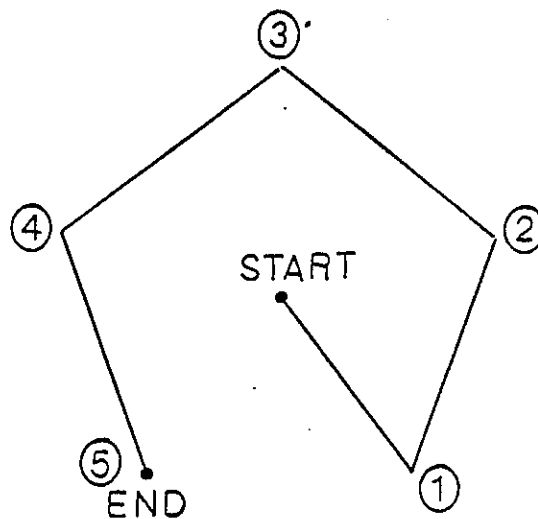


FIGURE 9.5.2 - BOLT HOLE PATTERN 5 HOLES

SECTION X

SUBROUTINES AND LOOPS

10.1 PROGRAM ENTRY OF SUBROUTINES

The subroutine feature gives the control a tremendous amount of versatility, extends its capability, simplifies and saves programming time. Subroutines must be written in Program #9 with the G7XX code appearing in the first block of the subroutine. Any other information entered in the same block with the G7XX code will be ignored by the control. Although Program #9, like other programs, must be written in consecutive blocks beginning with 9001, the subroutines themselves can be written in any order. Therefore, if subroutine 05 ends at block 9027, the next subroutine, say 01, must begin at block 9028. With these few simple exceptions, subroutines are written in the same manner that an ordinary program is written.

10.2 USING SUBROUTINES AS FIXTURE OFFSETS

One very important fact should be remembered when using subroutines.

When a subroutine is called, the part Floating Zero is temporarily shifted to the present axes positions. When a subroutine is completed, the main program is continued. As soon as the main program is reentered, the Floating Zero is shifted back to its original position.

This means that when a pattern is repeated more than once in a program, the pattern should be written as a subroutine. Then at any time, the machine can be positioned to the beginning of the pattern and the subroutine called. This will cause the subroutine (pattern) to be run. The subroutine may be called up as often as desired.

For example, if two vises were located on the machine, the machine could be positioned to the program zero of vise 1 and the subroutine called. When the first part in vise 1 was finished, the machine could then be positioned to program zero of vise 2 and the subroutine called. This will cause the second part (in vise 2) to be run. This procedure may be repeated as often as desired.

10.3 INITIATING AND LOOPING SUBROUTINES

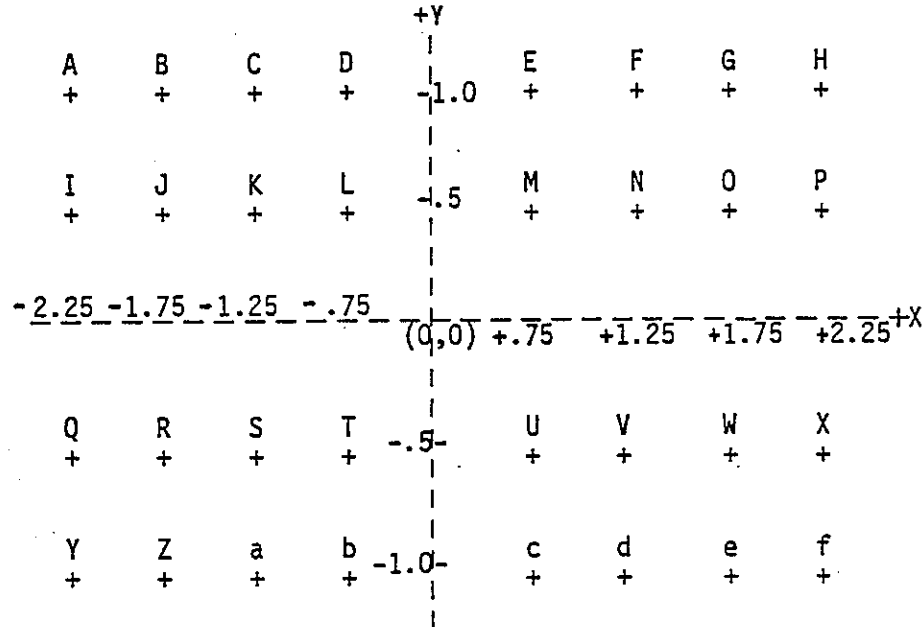
When using subroutines it is very important to know the following information:

1. Subroutines are programmed from the point of entry. In other words, the machine position at the time a subroutine is called up, acts as a Floating Zero for that subroutine and any position in the subroutine must be referenced to this point.
2. Although the point of entry is the reference point for a subroutine, the original Floating Zero is not removed and will remain the reference point when returning to the main program.
3. The G8XX code is used to call up and run a subroutine. The "X"s indicate the subroutine number. When a G8XX code is encountered in the program, the program is exited to Program #9, where the corresponding G7XX code is located and that subroutine is run in its entirety, including any subroutines called for from this subroutine. The second subroutine can call a third subroutine, which can call a fourth. This process can continue for 10 levels.
4. Subroutines can also be repeated or looped a programmed number of times by using the G8XX, LYY code (L and M are accomplished by the same key on the keypad). This commands the subroutine indicated by the G8XX code to be run "YY" times.
5. Nesting subroutines is accomplished by calling up another subroutine from a subroutine. This is done in the same way and by using the same codes as in the main program.
6. There is a maximum number of 99 different subroutines that can be programmed (701-799).
7. The maximum number of times a subroutine can be repeated or looped is 999 (L001-L999).
8. There is a maximum number of ten levels of nested subroutines. A nested subroutine is a subroutine which is called by another subroutine.
9. It is very important to TURN OFF cutter compensation (G40) prior to calling a subroutine (G8XX) or terminating a subroutine (G7XX), and remembering to turn it back on upon entry into the subroutine or when returning to the main program. This also applies to any X or Y offsets that may be active.

The fact that the original Floating Zero is maintained while running subroutines and that the subroutines can be programmed in the absolute or incremental mode from the point of entry, gives the control's subroutine feature versatility as well as making it easy to use. The looping capability allows short, simple programs to accomplish lengthy hole drilling patterns and other symmetrical patterns. The ability to nest subroutines offers several levels of power and added versatility to this feature.

EXAMPLE 10.1 SYMMETRIC HOLE DRILLING PATTERN USING SUBROUTINES

TOOL #4 (1/8"drill)



BLOCK #	LINE ENTRY INFO.
1001	F5, G0, G104, S1400, M3, M7, G81, P-.5
1002	G300, P.01
1003	X-2.25, Y1
1004	G801, L3, X.75
1005	G801, L3, X-2.25, Y.5
1006	G801, L3, X.75
1007	G801, L3, X-2.25, Y-.5
1008	G801, L3, X.75
1009	G801, L3, X-2.25, Y-1
1010	G801, L3, X.75
1011	G801, L3, M5, M9, G80, G98

9001	G701
9002	X0.5
9003	G701

Once the Floating Zero (at the center of the part) is set for this part the machine is ready to run this program.

Line 001 - of the program turns the spindle and coolant ON, activates the drill canned cycle for a depth of -0.5 inch, sets a feedrate of 5 IPM for feeding the drill down, and selects rapid feed for moving from one hole to the next.

Line 002 - Sets the R-Plane for this part.

Line 003 - positions the tool to hole A and drills the hole.

Line 004 - calls subroutine 01 three times which moves the X-axis $\frac{1}{2}$ inch to the right each time and drills a hole, thus doing holes B, C and D. It then moves the tool to position E by commanding X-axis to +.75 inch (from Floating Zero).

Line 005 - again calls subroutine 01 three times to do holes F, G, H and moves to hole I.

Line 006 - calls subroutine 01 three times to do holes J, K, L and moves to hole M.

Line 007 - does holes N, O, P and Q.

Line 008 - does holes R, S, T and U.

Line 009 - does holes V, W, X and Y.

Line 010 - does holes Z, a, b and c.

Line 011 - does holes d, e and f, shuts off the spindle, coolant, and drill canned cycle, and positions all three axes to their Floating Zero position. This program drills 32 holes with 14 lines of programming.

C A U T I O N S :

1. Make sure all tool offsets are de-activated (G100) on the move to the start point of a subroutine. Once the machine is in the correct position at the start point of the subroutine, the tool offsets can be re-activated (G1XX). If this procedure is not followed each time the subroutine is run its position relative to the rest of the part will progress, by the value of the offsets.
2. The move to the start point of a subroutine must not be done with cutter radius compensation active (G41, G42). The move should always be made in the G40 mode. If this procedure is not followed each time the subroutine is run its position relative to the rest of the part will progress by the cutter radius value.
3. Cutter compensation must be OFF (G40) when entering or leaving a subroutine.
4. When possible activate tool offsets (G1XX) in the subroutine so that other tool offsets will not cause cutter comp. error codes.
5. When using subroutines that cannot specify tool offsets then follow G1XX codes with a tool change command (M06) even if the tool is not being changed. This will insure that the subroutine is cutter compensated for the correct tool number.

SECTION XI

SWITCHABLE PLANE PROGRAMMING

11.1 PROGRAMMING ARCS IN THE XZ PLANE

The rules for programming an arc or circle remain essentially the same no matter which plane in which it is programmed. Only the G-code for plane of operation differs. The G18 code is used for the XZ plane, select the G02 code for the clockwise direction command, and the G03 code for the counterclockwise direction command.

Here, the clockwise and counterclockwise directions are determined by standing in front of and facing the machine. Of course, the starting point, center point, and end point for the arc will now be given as X/I and Z/K dimensions. It is still required to have a linear move preceeding and succeeding a circular move. Autoroutines are inoperable in XZ and YZ planes. It should also be noted that cutter compensation is inactive in the XZ and YZ planes.

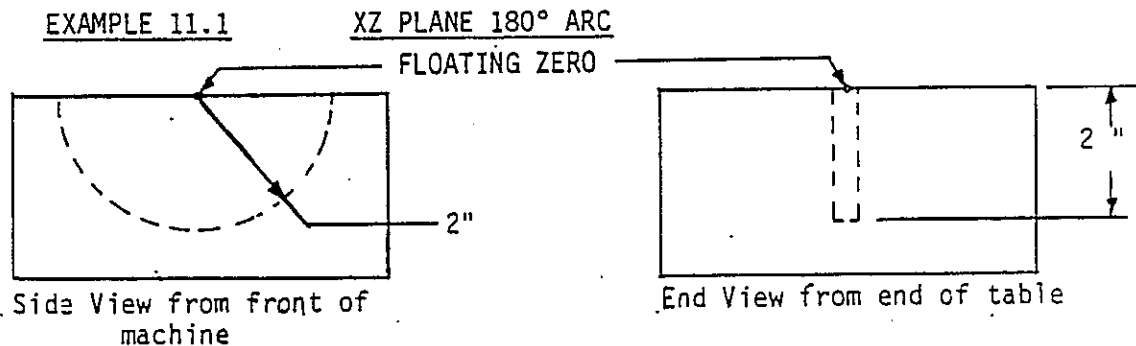


FIGURE 11.1.1 - XZ PLANE 180° ARC

Program # : 4 XZ Plane circular slot
Tool # : 5 (1/2" ballnose mill)

<u>BLOCK #</u>	<u>BLOCK LINE INFO</u>
4001	G105, M06
4002	G300, P.1
4003	G18
4004	F6, G0, G99, X2., Y0.
4005	S1200, M3, M7, G1, X2., Z0
4006	G02, R2, I0, K0, X-2., Z0
4007	G99, M5, M9, G100, G0
4008	G98, G17

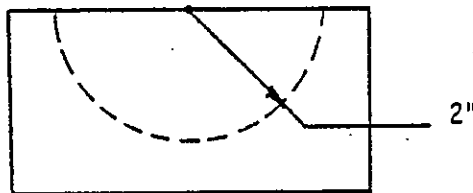
Block 001 Activates Tool Offset 5
 Block 002 Sets the R-Plane at .1
 Block 003 Selects the XZ Plane
 Block 004 Rapids Z to the R-plane and positions XY
 Block 005 Starts the spindle and coolant, and positions the tool to the start point of the slot
 Block 006 Defines and accomplishes a 180° arc with a 2 inch radius in the clockwise direction
 Block 007 and
 Block 008 Shuts OFF the spindle and coolant, and rapids the tool away from the part

11.2 PROGRAMMING ARCS IN THE YZ PLANE

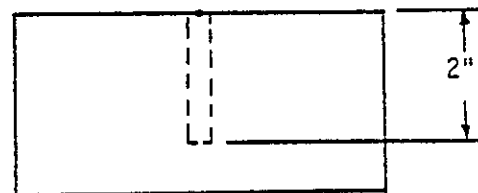
The only real change for programming an arc in the YZ plane is again the G19 plane select code. For a clockwise direction the G02 code is used, and the counterclockwise direction uses the G03 code. Again the start, center and end points for an arc will be given by J (Y) and K (Z) dimensions. Here the clockwise and counterclockwise directions are determined by standing at the right end of the table and looking down the length of the table.

EXAMPLE 11.2.1 YZ PLANE 180° ARC

FLOATING ZERO



Side View from Right Side
Machine



View from Front of Machine

FIGURE 11.2.1 YZ PLANE CIRCULAR SLOT

PROGRAM #: 5 YZ Plane Circular Slot
 TOOL #: 5

BLOCK #	BLOCK ENTRY INFO
5001	G105, M06
5002	G300, P.1
5003	G19
5004	F6, G0, G99, X0, Y-2.
5005	S1200, M3, M7, G1, Y-2., Z0
5006	G03, R2., J0, K0, Y2., Z0
5007	G99, M5, M9, G100, G0
5008	G98, G17

Block 001 Activates tool offset #5
Block 002 Sets the R-Plane at .1
Block 003 Selects the YZ Plane
Block 004 Selects a feedrate of 6 ipm, and positions the tool
just above the start point
Block 005 Turns ON the spindle and coolant, and moves the tool
to the starting point
Block 006 Accomplishes a 180° arc with a radius of 2 inches in
the counterclockwise direction
Block 007
and
Block 008 Turns OFF the spindle and coolant, and rapids the tool
away from the part

SECTION XII

USING THE HELICAL PROGRAMMING FEATURE

12.1 WHAT IS HELICAL?

As applied to the Centurion IV control "HELICAL" refers to a three dimensional spiral such as a screw thread. What it means to the programmer is that you can program the third axis linearly while the other two axes are defining an arc.

Programming the Helical move is very simple. Just program an arc and at the end of the arc statement program the third axis to some dimension. When the program is run it will cause the third axis to make a linear move to the dimension programmed while the other two axes complete the arc.

12.2 HELICAL PROGRAMMING EXAMPLES

EXAMPLE 12.2.1

HELICAL PROGRAM

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
----------------	-------------------------

1001	F20, G0, G17, X1, Y0, Z0
1002	G1, G2, R1, X0, Y0, X-1, Y0, Z-1
1003	G0, X0, Z0

Block 001 Causes rapid move to start of 180° arc

Block 002 Commands the XY axes to describe a 180° arc in the XY plane while Z makes a 1 inch move in the negative direction

Block 003 Returns the X and Z axes to their floating zero location

EXAMPLE 12.2.2

MULTI-HELICAL PROGRAM

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
----------------	-------------------------

2001	G10, G101, S1500, M3, G17, G300, P.2
2002	G99, X-.5, Y0
2003	F20, G1, Z-.01
2004	G3, R.5, X0, Y0, X-.3536, Y-.3536, Z-.2
2005	R.5, X0, Y0, X0, Y-.5, Z-.01
2006	R.5, X0, Y0, X.3536, Y-.3536, Z-.2
2007	R.5, X0, Y0, X.5, Y0, Z-.01
2008	R.5, X0, Y0, X.3536, Y.3536, Z-.2
2009	R.5, X0, Y0, X0, Y.5, Z-.01
2010	R.5, X0, Y0, X-.3536, Y.3536, Z-.2
2011	R.5, X0, Y0, X-.5, Y0, Z-.01
2012	M5, G98

Block 002 Position the axes to the start of the cut
 and
 Block 003
 Block 004 Causes X and Y to define a circle in 45° arc segments
 through with a radius of .5 inch. Through each arc segment the
 Block 011 Z axis is alternating between a .19 inch negative move
 and then returning to the start point during the next
 arc segment.

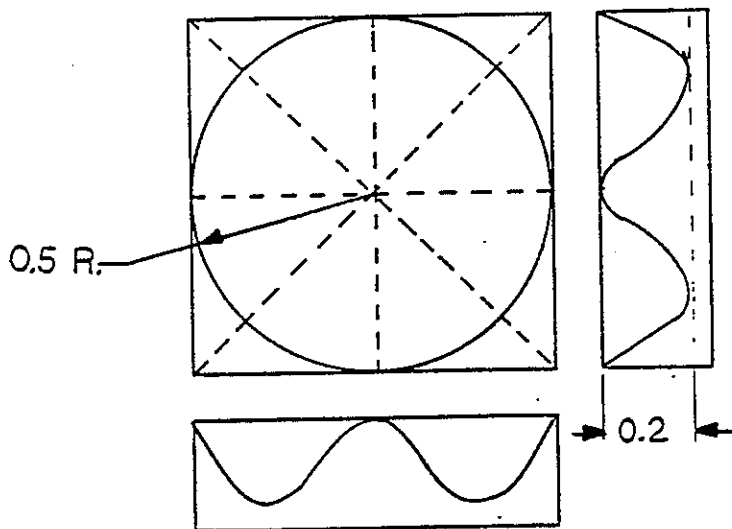


FIGURE 12.2.1 MULTI-HELICAL PART

Helical moves can also be made in the XZ and YZ planes. By putting a Helical program in a subroutine several loops of a spiral can be accomplished just by looping on the subroutine.

EXAMPLE 12.2.3 YZ HELICAL LOOPS

BLOCK #	BLOCK ENTRY INFO
3001	F2, G0, G101, G300, P.02
3002	S1000, M3, M8, G99, X0, Y0
3003	G1, G19, Z0
3004	G801, L5, G98, M5, M9, G17
9001	G701
9002	G2, R.5, J0, K-.5, Y0, Z0, X-.2
9003	G701

Block 001 Calls Tool #1 and sets the R-Plane
 Block 002 Starts the spindle and positions X and Y to start point
 Block 003 Feeds Z down to the start point and calls the YZ plane
 Block 004 Loops on the subroutine 5 times which causes the tool
 to describe 5 full loops while moving the X-axis -.2
 inches per loop. It then shuts spindle and coolant off,
 rapids to Floating Zero and calls the XY plane.

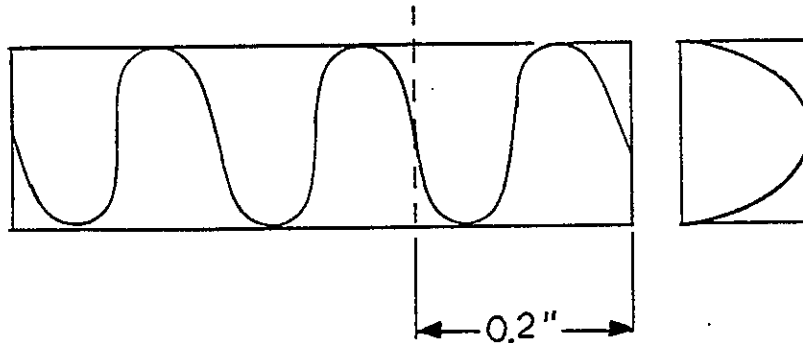


FIGURE 12.2.2 YZ HELICAL LOOPS

Note: Cutter compensation cannot be used with a Helical move.

SECTION XIII

CUTTER RADIUS COMPENSATION

13.1 INTRODUCTION

This section will explain how the cutter compensation works and give pointers on how to use optimally.

Cutter compensatin is the displacement of the tool path, normal to the programmed path by the amount equal to the cutter radius. The programmed path can be figured by the programmer for a zero tool radius.

If the parts program is written for a zero tool radius, ie: directly off the print, then by entering the actual tool radius into the system and activating cutter compensation, the operator can make the control calculate the displaced path.

The operating system compensates the whole parts program before the program is ever run. This feature, allows most critical programming errors to be detected by the system before actually running the program. This can be a big time saver. If an error is detected, the system puts the correct error code and the Block Number at which the error was detected on the CRT. If more than one error is detected in the program, the successive errors will only show up on the display after the previous ones are fixed by the operator.

In some cases where the operator has entered incorrect end points for arc, the compensation, if turned "ON", will fill in the correct end point and not give an error code.

This cutter compensation is full look-ahead compensation. Throughout the program, the control keeps a record of the previous programmed point, the current programmed point, and the next programmed point along the tool path.

These three points, information on how they are connected, the cutter radius, and whether it is a left or a right cut, are absolute requirements by the control to calculate the current compensated point.

After each successful calculation of a compensated point, the current programmed point becomes the previous programmed point, the next programmed point becomes the current programmed point and a new programmed point is read up to become the next programmed point. This mechanism is repeated over and over again until the end of the program is reached. This sequence should be understood clearly in order to understand many points that will later come up on how the compensation works.

This compensation is truly intersectional. Given these three points, the control calculates the intersection of the compensated path between the previous and the current programmed points and the compensated path between the current and the next programmed points. These paths can be a mixture of straight lines and arcs.

Because of the intersectional nature of the compensation package, there has to be an intersection of all the displaced paths for the system to work. If there is no intersection between two paths, the control will give an error code and will not even begin to execute the parts program.

NOTE: All cutter compensation examples are shown without Z-axis moves.

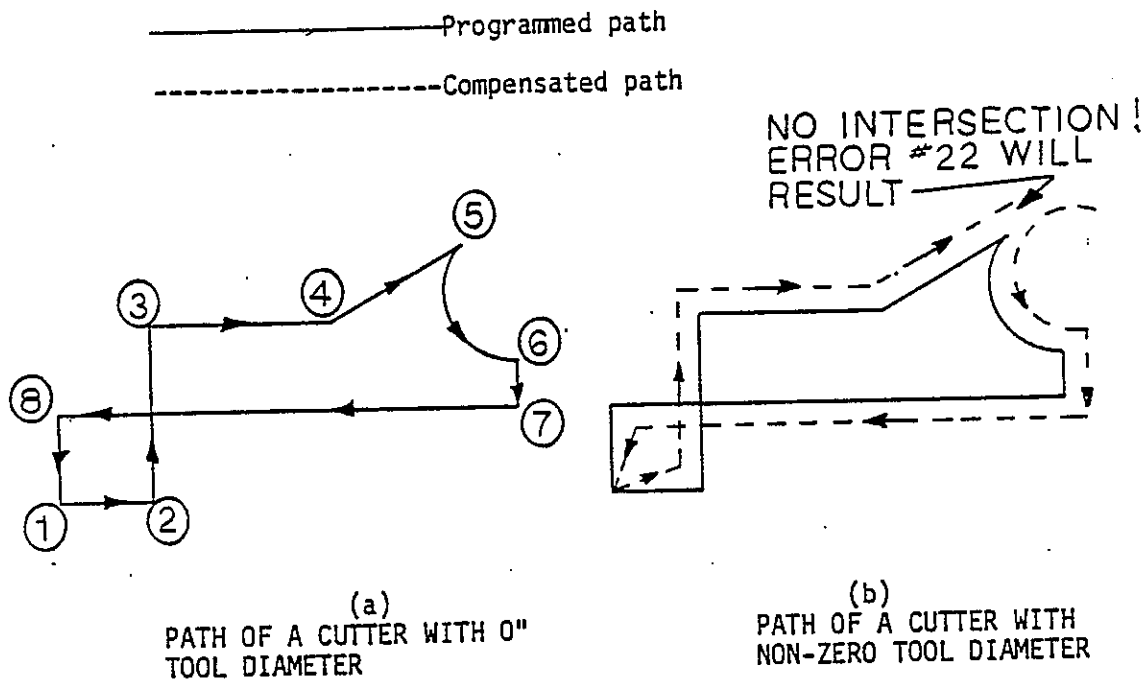


Figure 13.1 EXPLANATION OF HOW DISPLACED TOOL PATHS CANNOT HAVE AN INTERSECTION

The solution of the above part is to introduce a 0.00001" move at Point 5 between the non-intersecting surfaces.

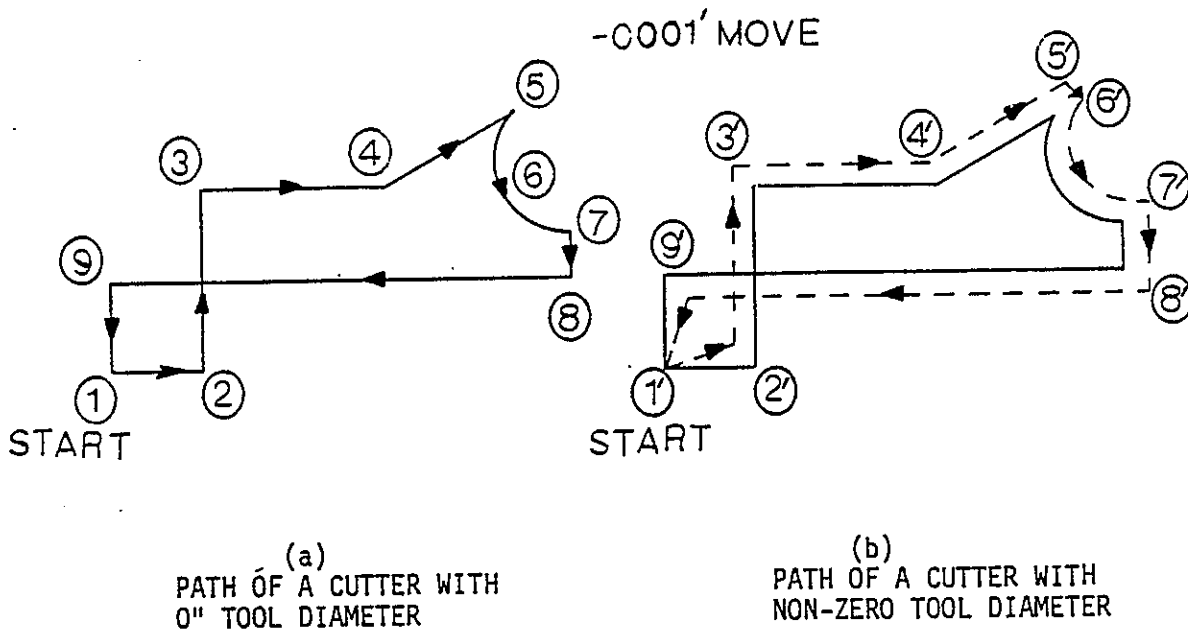


Figure 13.1.2 EXPLANATION OF HOW A .0001" MOVE SHOULD BE INTRODUCED
TO SOLVE A NO-INTERSECTION PROBLEM

In some cases, the system will find an intersection but it will be unreasonably far away from the part. Again, in such cases, a .0001" move can be inserted to solve the problem.

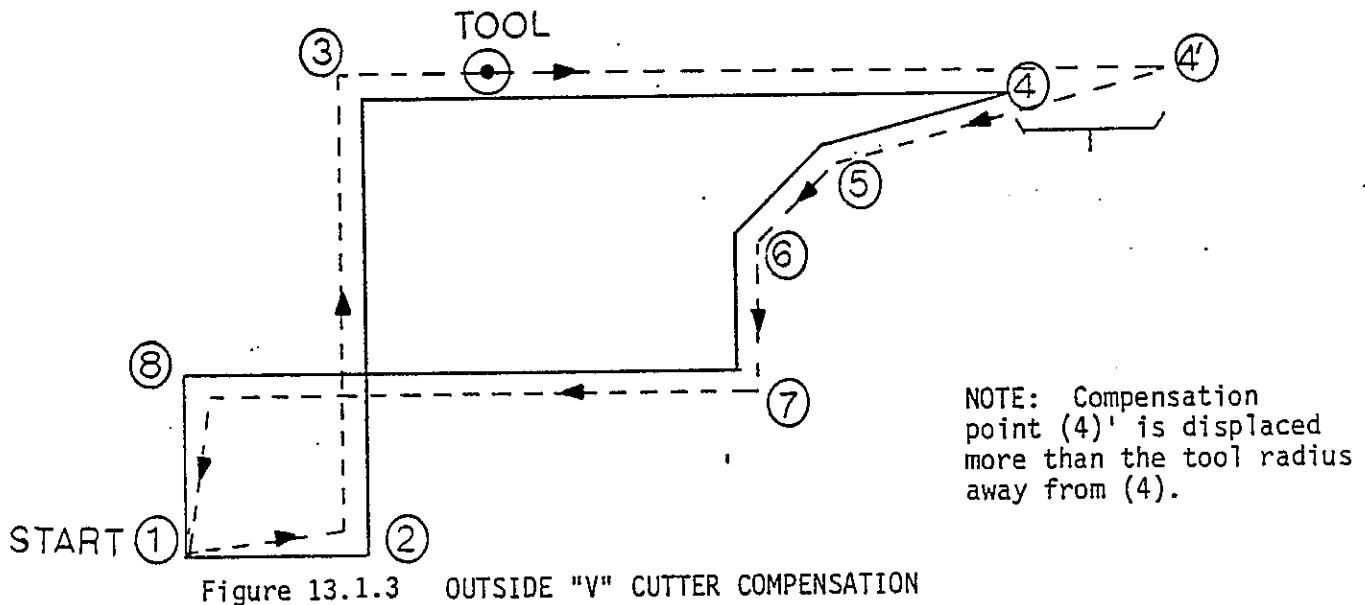


Figure 13.1.4 shows how a .0001" move added at point 4 has saved an unnecessary departure.

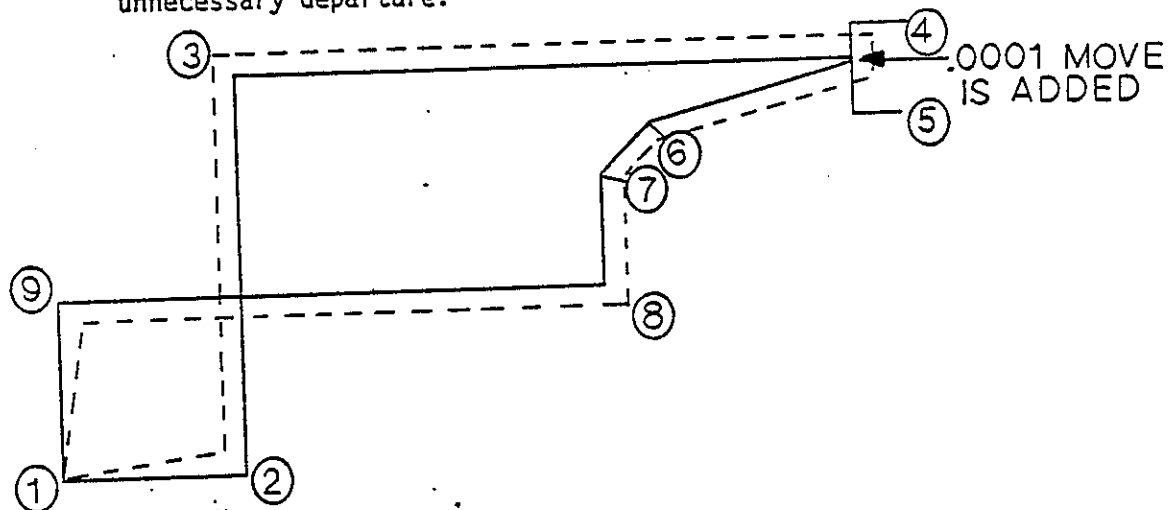


Figure 13.1.4

OUTSIDE "V" CUTTER COMPENSATION SOLUTION

The following case shows how the compensated point for an inside "V" will stay away from the programmed point by more than the tool radius.

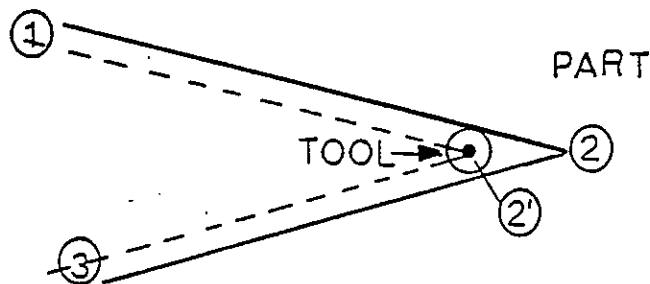


Figure 13.1.5

INSIDE "V" CUTTER COMPENSATION

NOTE: The tool stays away from the programmed point (2) by a distance more than the tool radius.

If the compensated point (2) was any closer to (2), the tool would gouge the sides of the part.

13.2 SAMPLE PART EXERCISE

As the system requires 3 points to generate a compensated point, care should be taken when the cutter compensation is turned on or off.

The magnitude for the axis movement for leaving or entering the part has to be greater than the amount of cutter compensation expected. Looking at Figure 13.21 it will be seen that the machine position at point 1 is identical for both the compensated as well as the uncompensated tool path.

This is because the first point cannot be compensated as it has no previous point to make up a 3 point set for the compensation routine. Therefore, the compensation should be turned on before the tool enters the work.

For ease of programming, the tool should enter and leave the part perpendicular to the part surface. This is not a strict requirement, but simplifies understanding how the cutter compensation will behave entering and exiting the workpiece. If in doubt on how the displaced tool path will look, it is advisable to rough sketch by hand, lines parallel to the part surface from start to end. Many times problems will become apparent right away.

EXAMPLE 13.2.1 HOW TO DETERMINE WHAT THE COMPENSATED PATH WILL LOOK LIKE

Step 1 Sketch actual part and label points in order of sequence

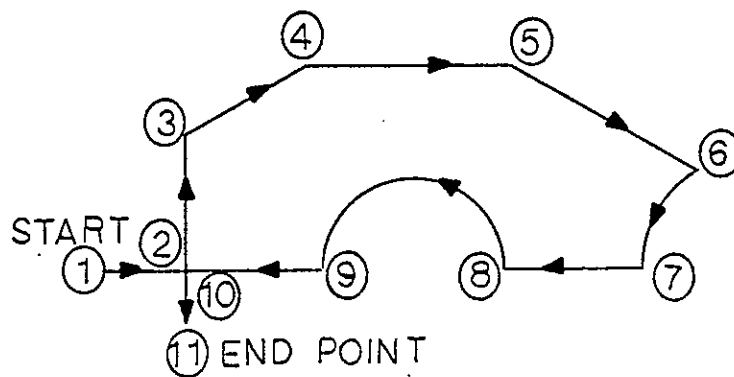


Figure 13.2.1 STEP 1

Step 2 Sketch lines displaced tool radius away from part surface from point (1) to point (11)

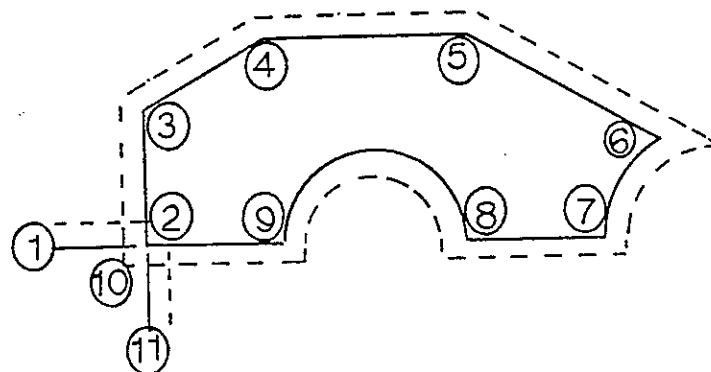


Figure 13.2.2 STEP 2

Step 3

Check if all paths in sequence intersect. If yes, then except for the start and end point, connect the displaced path and label points of intersection. If even 1 intersection cannot be found, the part will not run if the error is not corrected.

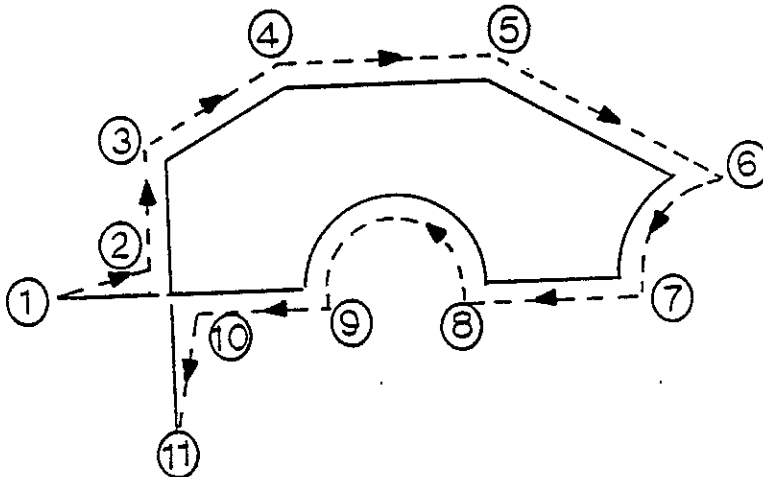


Figure 13.2.3

STEP 3

Step 4

Since points (1) and (11) do not have two points on either side, they will be the compensated points. Therefore, connect them to their neighboring compensated points to arrive at the actual displaced cutter path.

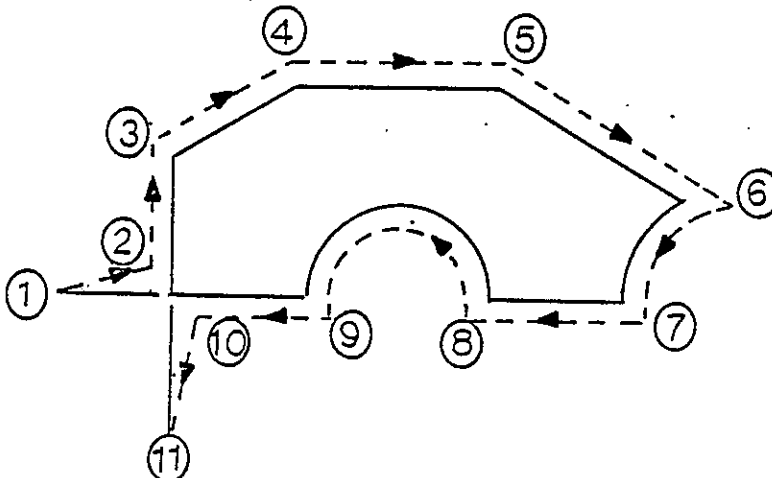


Figure 13.2.4

STEP 4

Step 5

The above displaced path is what the system will trace if the part is run. However, a problem has become apparent from the rough sketch.

Note that the lower left hand corner will be left uncut because of the fact that the tool in going from (1) to (2) will leave a little notch of uncut material. A similar case is obvious in the tool path from (10) to (11). There again the corner will be left uncut.

The solution is to rearrange the start and end points so that the corner is properly cut.

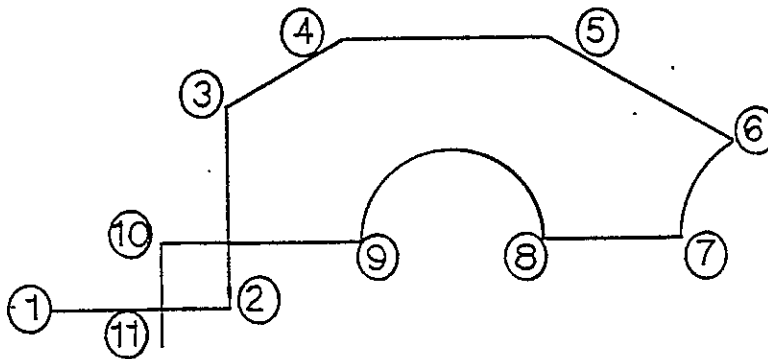


Figure 13.2.5

STEP 5

Step 6

Note how points (1) and (2) and (10) and (11) have been moved a little. The result will be as follows:

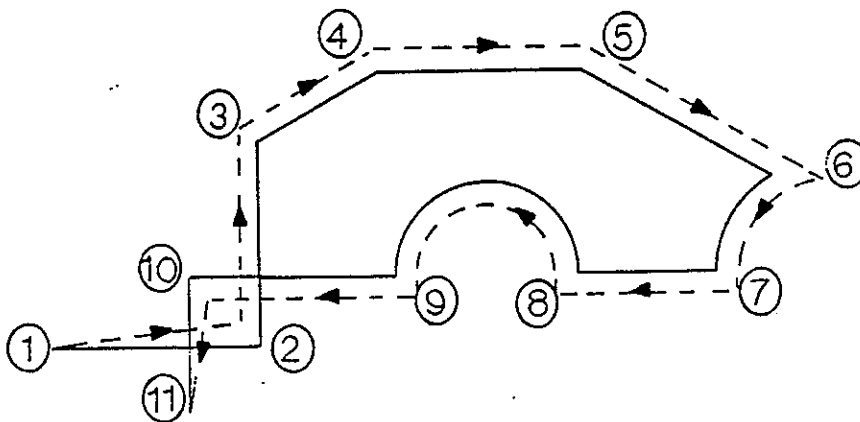


Figure 13.2.6

STEP 6

It is now seen that when the tool moved from (2) to (3) and (9) to (10), the corner will be properly cut.

In the above Example 13.2.1, the cutter compensation was turned on by using the G41 command for turning on left cutter compensation at point (2) and turned off by using the G40 command at point (11).

A few more points are to be noted for the above example.

1. Points (1) and (11), being the start and end points were chosen so that they lie sufficiently away from the part surface. "Sufficient" is a distance more than the total expected compensation.
2. Circle at (9) and (8), points to complications that will arise if the tool radius is increased indefinitely. As the radius is increased (9) and (8) will keep moving closer to one another. For some value of the radius, they will become identical. If the radius is increased further the tool radius will have become too large to make that circle and the system will give an error (E22) telling the operator that an intersection cannot be found at that line.
3. If a compensated path can be successfully sketched by hand, then it will run on the system. However, if the sketch yields a missing intersection, the control will give an error (E22 and E23).
4. Until the operator becomes familiar with cutter compensation, it is advisable that rough sketches be made for the compensated path before the part is run as a program.

13.3 . TOOL MANIPULATION

The following will explain how to manipulate the compensated points for bringing the tool up and down.

Assume that the tool is required to come down to mill depth exactly at the lower left corner. Also, it is to be brought up after it has completed milling the part again at the lower left corner.

Looking at Figure 13.2.6, we can see that the tool is on the correct cutter path at point (2) and the next programmed point is at (3). In this Example 13.1, the cutter would have to be brought down at point (2). Similarly, at the end of the program, point 10 to bring the tool up.

This could be troublesome, especially if there were obstructions around the corner.

In similar cases, the tool can be made to go down and up at the exact desired points by the following simple manipulation, see Figure 13.3.1.

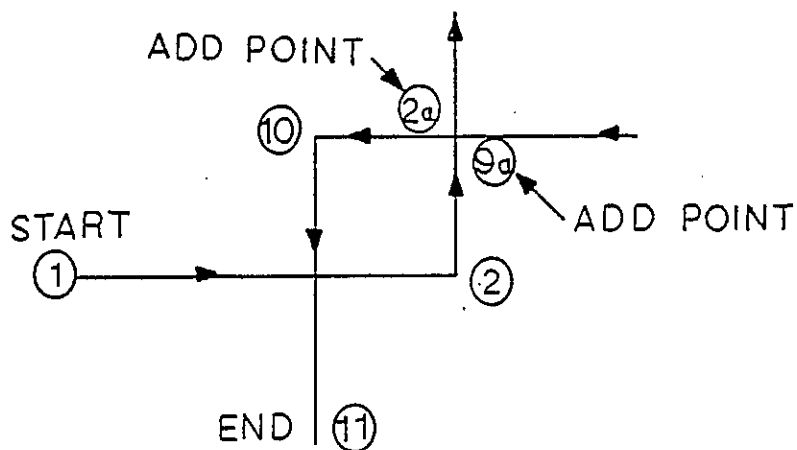


Figure 13.3.1

SIMPLE MANIPULATION

The solution is the addition of a point (2a) which is co-linear with points (2) and (3) and a point (9a) which is co-linear with points (9) and (10).

The tool can now be brought down at point (2a) and brought back up at point (9a).

EXAMPLE 13.3.1

If the Part "A" was a cavity, then the start and end point would have to change. Simply changing the G41 to G42 (right cut) will not help. This is because the tool would still come down on point (2a) at the beginning. This would be disastrous for the cavity because in doing so the tool would cut into the side (9) to (9a). The reason for that is that as explained earlier, the system uses the previous, current, and the next programmed points to calculate the current compensated point. Therefore, in the case of point (2a), only lines (2) to (2a) to (3) are used to calculate its compensation. This procedure does not include line (9) to (9a).

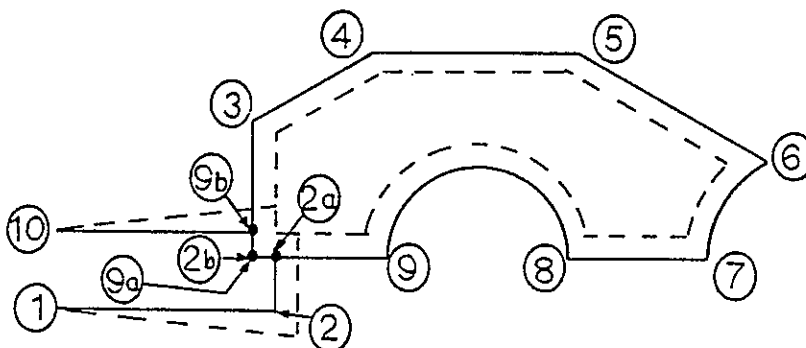


Figure 13.3.2

ADDITION OF POINTS (2b) AND (9b) FOR A CAVITY

In Figure 13.32 the tool should be brought down at points (2b) and taken back up at (9a).

The purpose for adding line (2a) to (2b) is to give advance information to the system about line (9) to (9a), which happens some 8 blocks later. The length of (2a) to (2b) can be as small as .0001". It is important to note that for the correct advance information, the slope of (2a) to (2b) has to be the same as the slope of (9) to (9a). In this case, the slope is zero.

Similarly, the purpose for adding lines (9a) to (9b) is to give past information to the system about line (2b) to (3), which happened some 8 blocks earlier. Again, the slope of (9a) to (9b) has to be the same as the slope of (2b) to (3), in this case infinity.

Regardless of the cutter radius, the tool can now always be brought down on the corner at (2b) and brought back up at (9a).

Note that the start and end points can either be made to lie inside or outside the cavity.

EXAMPLE 13.3.2

PROGRAMMING WITH CUTTER COMPENSATION

When programming with cutter radius compensation, the first and last move the cutter makes should be done off of the part per Figure 13.3.3. The movement made prior to cutting should be at least a distance of the cutter diameter being used.

BLOCK #	BLOCK ENTRY INFO
1001	G0, X-1., Y-1.
1002	G41, X0
1003	G1, Y3., F10
1004	X3.5
1005	G3, R.5, X4.,Y3.,X4.5, Y3.
1006	G1, X6.8
1007	Y2.
1008	X4.5, Y0
1009	X-1
1010	G40, Y-.5

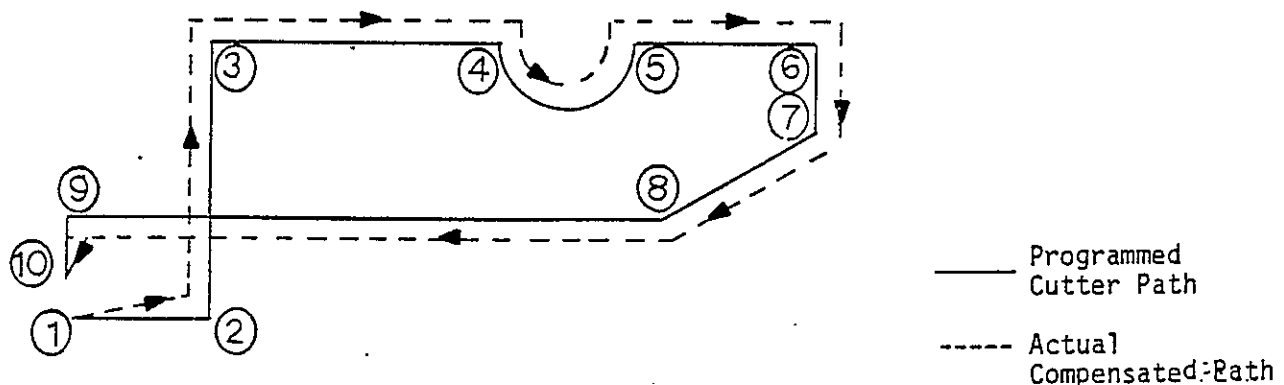
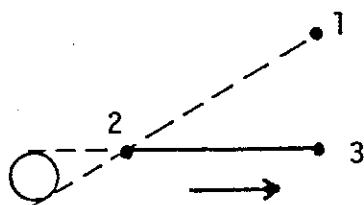
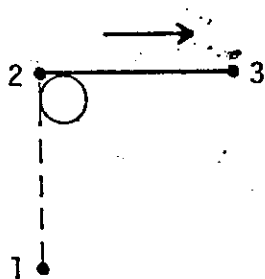
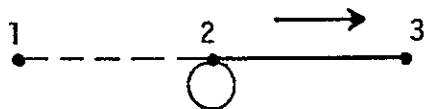
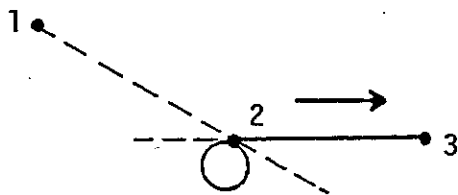
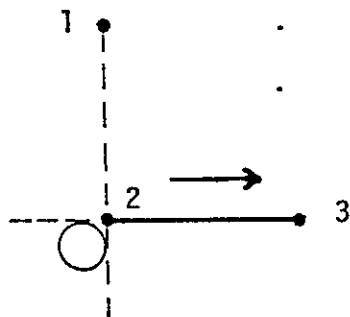


Figure 13.3.3 PROGRAMMING WITH CUTTER COMPENSATION

TOOL RIGHT CUTTER COMPENSATION

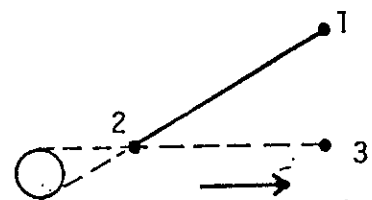
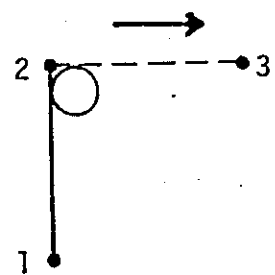
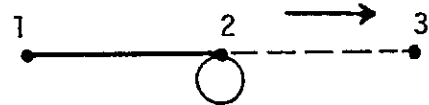
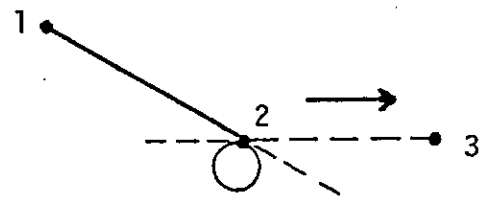
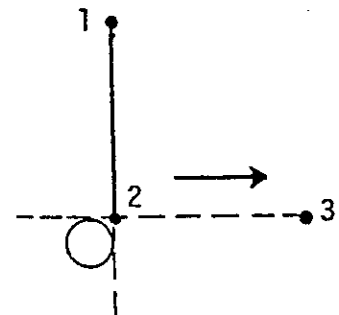
PIERCE

- 1 = point on part before pierce point
- 2 = pierce point
- 3 = first cut move



RETRACT

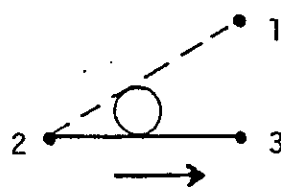
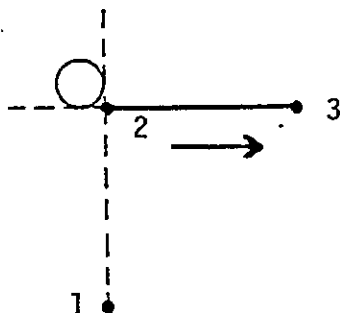
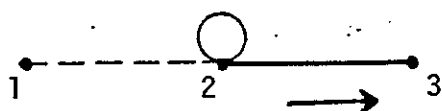
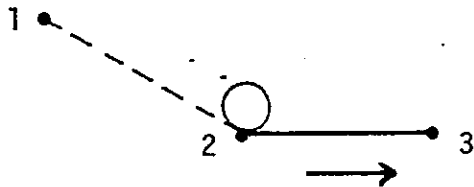
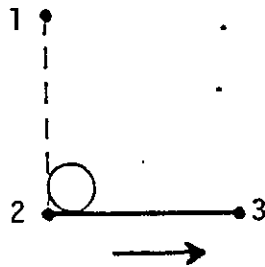
- 1 = last position before retract
- 2 = tool retract position
- 3 = point after retract



TOOL LEFT CUTTER COMPENSATION

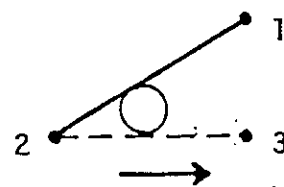
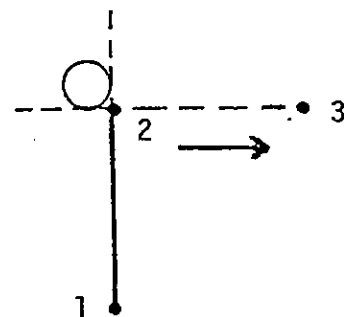
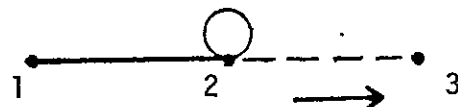
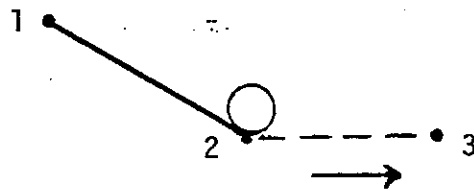
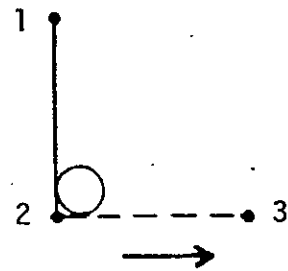
PIERCE

- 1 = point on part before pierce point
- 2 = pierce point
- 3 = first cut move



RETRACT

- 1 = last position before retract
- 2 = tool retract position
- 3 = point after retract



13.3.1 NOTES ON CUTTER COMPENSATION

- ** Cutter Compensation cannot be turned off at the end of a circular move.
- ** Turning compensation on can be done both in a block with no axis move, and in a block containing axis moves.
- ** There is no restriction on how many successive blocks can have no axis information.
- ** There is no restriction on how lines enter and leave arcs. They can have any angle of intersection as long as an intersection exists.
- ** The only restriction on arcs intersecting arcs is that they cannot be concentric (ie: an arc segment cannot be broken into two moves).
- ** Given the correct centers and radii for two intersecting circles, the system automatically checks and corrects the programmed point of intersection when compensation is turned on.
- ** It was explained in Section 9 on Autoroutines, that all the pocket and bolt circle routines use the present axis position as their start point. For this reason, it should be made sure that the cutter compensation is turned off in a program using these routines, so that the axis can position to the programmed center. If the compensation center is used, the whole pocket will be shifted.
- ** If the programmed point rather than the compensated point is desired, a G40 command should be added to the block containing that point. The G40 should have been a part of the previous block but if that block had axis information, the previous point would not be compensated.
- ** Similarly, if the point after the desired programmed point is to be compensated, a G41, or G42 should be added to the block containing that next point or to any non-axis command block between the two.
- ** While the system is processing the parts program, the CRT display will indicate cutter compensation in progress indicating that the system is in the busy mode. The program execution can only commence after the cutter compensation procedure is complete.
- ** A program cannot end with a circular arc move. If the last line has to be an arc, a ".0001" linear move should be used for program termination.

13.3.2 FINAL NOTE

Roughing and finishing passes can be easily made by first entering a tool radius value larger than the actual measured tool radius by the amount of stock to be left on the part for the finish pass.

When the program is run, the resultant part will be oversize. Now by entering the actual cutter radius into the system and running the program once more, the finished part size will be obtained.

Thus, by entering a larger or smaller tool radius, the part can be made under or over-size.

13.4 "TRIG-HELP" PROGRAMMING AID PACKAGE

This section describes the "TRIG-HELP" feature of the control.

As the name "TRIG-HELP" implies, this unique feature aids the programmer in writing his parts program by automatically doing various trigonometric calculations that are required to find start and end points or arcs.

This feature is aimed at reducing offline programming time and puts the power of an offline system into the control.

In figure 13.4.1, the points (2) and (3) are very easy to find. On the other hand, in Figure 13.4.2, the solution is not so trivial. It needs calculations that would require a calculator.

The "TRIG-HELPer" will find points (2) and (3) automatically. As the estimated points, the programmer need only give those so that they lie on the lines (1) to (2) and (3) and (4).

In other words, the start point of the circle can be specified to lie anywhere along the part (1) to (2). Similarly, the end point of the circle can be specified to be anywhere along path (3) and (4).

With Cutter Compensation turned on, and a tool radius of zero, the program will yield the correct path.

EXAMPLE 13.4.1

TRIG-HELP

ESTIMATED PROGRAM

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>
1001	G0, X0.2, Y0, G109
1002	G1, G41, X0.4, Y0.2
1003	G2, R.5, X1.5, Y1, X2.6, Y0.2
1004	G1, X2.8, Y0
1005	G100

Estimated start
point Circle w/
estimated end
point

PROGRAM WILL RUN AS FOLLOWS:

1001	G0, X0.2, Y0, G109
1002	G1, G41, X1.0298, Y00.8298
1003	G2, R.5, X1.5, Y1, X1.9702, Y0.8298
1004	G1, X2.8, Y0

Exact start point
substituted by control.
Exact end point
substituted by control.

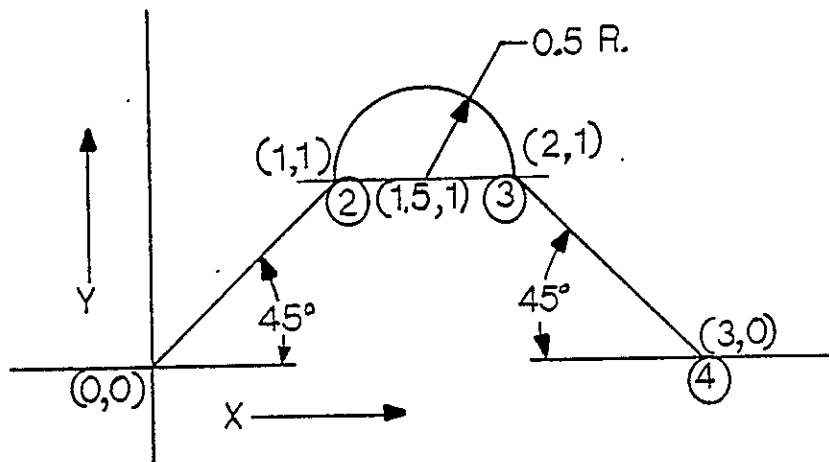


Figure 13.4.1 POINTS (2) AND (3) EASY TO FIND

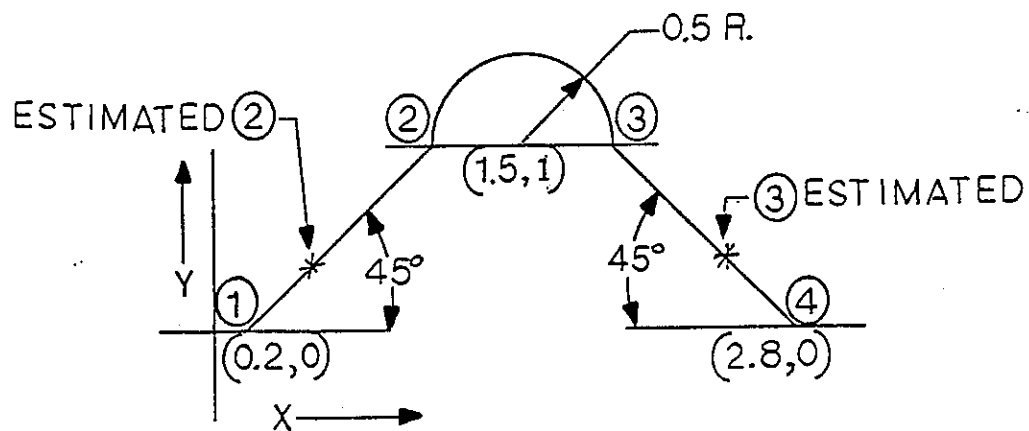


Figure 13.4.2 "TRIG-HELP" REQUIRED TO FIND POINTS (2) AND (3)

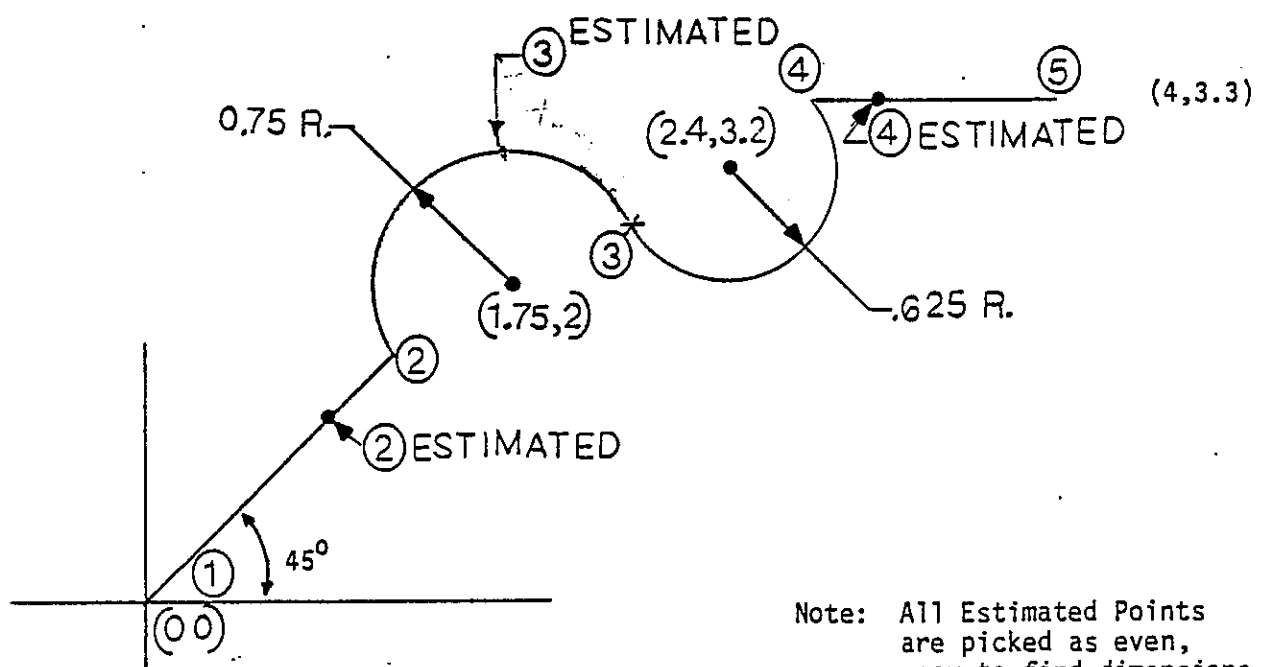


Figure 13.4.3

ESTIMATED POINTS

Figure 13.4.3 shows how to estimate for all three cases of line to circle, circle to circle, and circle to line paths.

The program is written as follows:

EXAMPLE 13.4.2

"TRIG-HELP"

ESTIMATED PROGRAM

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO</u>	
1001	G0, X0, Y0, G109	Estimated start point of 1st circle, 1st circle definition, & estimated end point of 1st circle
1002	G1, G41, X1, Y1	
1003	G2, R.75, X1.75, Y2, X1.75, Y2.75	
1004	G3, R.625, X2.4, Y3.2, X3.5, Y3.3	2nd Circle definition and estimated end point of 2nd circle.
1005	G1, X4, Y3.3	
1006	G100, G40, G98	

PROGRAM WILL RUN AS FOLLOWS

1001	G0, X0, Y0, G109	Exact start point substituted by control. Exact end point substituted by control.
1002	G1, X2.3596, Y1.3596	
1003	G2, R.75, X1.75, Y2, X2.0316	
1004	G3, R.625, X2.4, Y3.2, X3.0169	Exact end point substituted by control.
1005	G1, X4.0, Y3.3	
1006	G100, G40, G98	

13.4.1 THINGS TO REMEMBER WHEN ESTIMATING POINTS

- ** Estimating should only be used with line/circle, circle/circle, and circle/line paths.
- ** For line/circle and circle/line, the start and end points estimates must lie on the line. ie: the slopes of the lines entering or leaving the arc must be correct.
- ** If two paths intersect at two points then the estimated point should be closer to the desired point of intersection.
- ** If the above points are met, there is no limit on how far the estimated point is away from the correct point.
- ** When estimating the intersection of one arc to another arc the end points chosen should be on the arc. The easiest point to pick on an arc is at one of the quadrant points (0°, 90°, 180°, or 270°)

SECTION XIV

FORMAT FOR PARAMETRICS

Any place where parametric block is desired a G-code that signified start of parametric block must appear alone on a line. G500. At the end of the parametric block a G501 must appear to signify its end.

A Parametric statement must be on a line by itself and be enclosed by a (, at the start and ,) at the end. Parametrics are referenced by PXX where XX=00 through 99 (ie: 100 total parametrics).

A parametric statement can be of the following form:
(A (,code delineats comments, A (, delineates parametric statement.)

(, P1 = 1.6 ,) This assigns value 1.6 to Parameter 1.

(, If $PIEQ = P2 + P3$ GOTO L1 ,), if $P = P2 + P3$ program flow will transfer line number having label L1.

(, GOTO L1 ,), This causes unconditional jump to label L1.

(, L1 ,), This labels line as L1.

P00 through P99 is valid range for P - numbers.

G500 is modal and is cancelled by G501. Any axis commands following G500 will be interpreted as parameter references; ie: after a G500 the line containing the statement X1, Y2 will assign value of parameter 1 to X and 2 to Y.

When the system comes across the command G500 it interprets that as a command

LIST OF OPERATORS

ARITHMATIC:

- * Multiply
- / Divide
- + Add
- Subtract
- SQRT Square Root
- [Left Bracket
-] Right Bracket
- = Assignment
- FRAC Will Return Fractional Part
- INT Will Return Integer Part
- ABS Will Return the Absolute Value

TRANCENDENTAL:

SIN Sine with argument in degrees
COS Cosine with argument in degrees
TAN Tangent with argument in degrees

INPUT

INPUT P1 will fetch value of P1 from Keyboard
INPUT"INPUT P1", P1 will print the message and
then fetch value of P1

Using the INPUT statement to read input channels:

The format for the statement is:

(INPUT, ISnn, Pmm)

(INPUT, IPnn, Pmm)

nn is the channel number. This is a base10 number which corresponds to the hexadecimal channel number marked on the system schematics. ie:

nn	0	1	2	 9	10	11	12	13	14	15	16	17.....
channel#	00	01	02	09	0A	0B	0C	0D	0E	0F	10	11.....

ISnn will read the status of single input channel nn.
IPnn will read the 4-bit value on Parallel channel nn.

Pmm is the parameter into which the channel value will be written. In the case of a single input channel (IS), Pmm will be a "1.0" if the channel is high, and "0.0" if it is low. Parallel input channels (IP) will cause a number between 0 and 15 to be written into Pmm; the number represents the 16 unique states of a 4 line (4 bit) parallel channel.

An example of the INPUT statement:

A clamp relay contact is to be sensed. If the clamp is open a message will be shown on the console display, and the program will wait for the clamp to close. The clamp is on single input channel 06.

```
1001      G500
1002      (,L1,)
1003      (, INPUT, IS06, P1,)
```

This is a label for the GOTO in block 1006. Parameter 1 will be loaded with the condition of single input channel 06: "1" if the clamp is closed and "0" if it is open.

```
1004      (,IF P1 EQ 1 GOTO L2,)
```

Skip blocks 1005 and 1006 if the clamp is closed.

```
1005      G4 F002.0 (Waiting for clamp)
```

This delays for 2 seconds. The comment will be displayed on the screen so that the operator can know why the program is waiting.

```
1006      (,GOTO L1,)
```

This will force a repeat of the previous three blocks. The control is "looping", as it waits for the clamp to be closed. Block 1004 will "jump" here when it finds the clamp closed (P1 = 1). Block 1008 will now be processed. The control is out of its input-test-delay loop.

```
1007      (,L2)
```

RELATIONAL:

EQ Equal
NE Not equal
LT Less than
GE Greater or equal

JUMP (BRANCH):

(, GOTO LXX,) Unconditional jump to label XX
(, IF <EXP1> OP1 <EXP2> GOTO LYY,) where <EXP1> and <EXP2> are expressions consisting of arithmetic operators

LABELS:

(,LXX,) where XX = 1-99. Labels should be at the beginning of a line. Label numbers should not be duplicated.

EXAMPLE:

SUBROUTINE TO CUT 1 60° FAN BLADE

```
9001 G701
9002 G500
9003 (,P7=.5,P5=2,P6=60,P8=.5,)
9004 (,P10=15,P11=P0=P17=0,)
9005 (,L1,)
9006 (,P9=.1*P11,P13=P8*SIN P6,P14=P8*COS P6,P16=P8,)
9007 (,P15=-P9,)
9008 X.0016 Y.0017 Z0
9009 G3 R.0008 XOY0 X.0013 Y.0014 Z.0015
9010 G2 R.0008 XOY0 X.0016 Y.0017 Z0
9011 X.0016
9012 (,P11=P11+1,P8=P8+.1,)
9013 (,IF P11 LT P10 GOTO L1,)
9014 G501
9015 Z.1
9016 XOY0
9017 G701
```

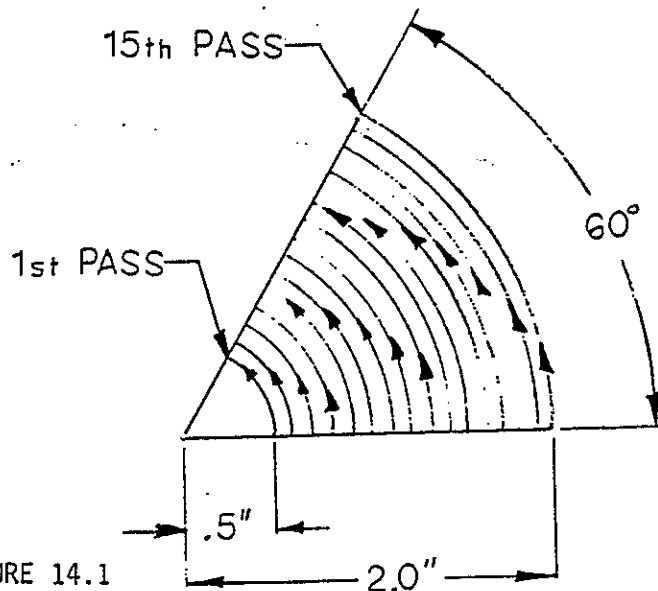


FIGURE 14.1

In the above example the tool makes 15 passes to make a blade 2" with Z going down 1.4" from one tip to the other. The blade sweeps 60°. This program can be modified and repeated six times to complete the fan.

P0 Z start point and end point of CW arc.
P5 is the outer radius of the blade
P6 is the angle it sweeps
P8 is the dynamic radius that increases from .5 and goes to P5
P9 is the dynamic Z depth that goes from 0 to 1.4"
P10 is the number of times for looping =15
P11 is the current # of loops goes from 0-15
P12
P13 X end point at end of CCW arc
P14 Y end point at end of CCW arc
P15 Z end point at end of CCW arc
P16 X end point at end of CW arc
P17 Y end point at end of CW arc

Looking at the program it is seen that line

9001 is just the normal subroutine header
9002 is the declaration of a parametric block the system will
 evaluate the entire block on encountering this command.
 The evaluation is done up to a G501 (end of parametric
 block) or a jump or branch statement.

In the case of this program the evaluation will be done up to line 9013,
which is a IF branch statement.

9003) These lines set up some initial parameters
9004)
9005 This is a Label for the command at 9013 to loop on

9006) These two lines are the actual calculations for the fan
9007) blade. They update parameters 9,13,14,15,16.

9008)
9009) These describe the arc and Z moves by using parameters
9010) 0,8,13,14,15,16,17,18
9011)

9012 This updates the loop count and radius of the arc
9013 This checks if 15 passes have not been done and loops back
 to 9005
9014 This terminates parametric block
9015 This brings Z up to .1
9016 this brings X, Y to 0
9017 End of Subroutine

Note by modifying 9006 and 9007 a little and introducing a parameter for rotation this program will cut a complete fan.

NOTES ON SYNTAX

<u>PARAMETER STATEMENT</u>	<u>SYNTAX</u>
(,P1=1+[P3*P5],)	OK
(,P1=1+(P3*P5),)	WRONG
(,P1=5, P2=6, P3=-1.6,)	OK
(,P1=5 P2=6 P3=-1.6,)	WRONG
(,P1=SIN [30+P1*P3],)	OK
(,P1=SIN 30,)	OK
(,P1=P5, L1,)	WRONG
(,L1, P1=5,)	OK
(,P1=1, IF P1 EQ1 GOTO L1,)	WRONG
(,IF P1 = 1 GOTO L1,)	OK
(,P1=SIN SQRT COS [30+P3],)	OK
(,P1=SIN [SQRT [COS [30+P3]]],)	OK
(,P1=P1+P3*[P6 EQ 6],)	OK

II. Parametrics

If the control is in the parametric mode (i.e. G500 instruction). axis moves are interpreted as follows:

A. Direct parametrics

1. (,P12 = 1.5,)
X 00.0012 (X moves to 1.5)
2. (,P12 = -1.5,)
X 00.0012 (X moves to -1.5)
3. (,P12 = 1.5,)
X-00.0012 (X moves to -1.5)
4. (,P12 = -1.5,)
X-00.0012 (X moves to 1.5)

B. Indirect parametrics

(If any digit other than the least significant two digits is non-zero, axis move is indirect.)

1. (,P12 = .0016,)
(,P16 = 1.5,)
X 00.0112 (X moves to 1.5)
(Same as X 00.0112 or X 99.9912)
2. (,P12 = .0016,)
(,P16 = -1.5,)
X 00.0112 (X moves to -1.5)
3. (,P12 = .0016,)
(,P16 = 1.5,)
X-00.0112 (X moves to -1.5)
4. (,P12 = .0016,)
(,P16 = -1.5,)
X-00.0112 (X moves to 1.5)
5. (,P12 = -.0016,)
(,P16 = 1.5,)
X 00.0112 (X moves to -1.5)
6. (,P12 = -.0016,)
(,P16 = -1.5,)
X 00.0112 (X moves to 1.5)

SECTION XV

SYSTEM PARAMETER SET-UP PAGE

On power up, if the system gives an E07 error, the system parameters have to be initialized before the machine can be operated. These machine parameters must be entered normally on read in via magnetic tape. If the system has battery backup, then, once read or entered the parameters stay intact. Go to the end of this section for directions on how to store and retrieve the system parameters from magnetic tape.

To enter the parameters the key sequence is:

DISP
STAT

9

CE

This will initialize the table for entry.

To call up a specific parameter follow the initialize routine with "DISPLAY STATUS", "9" "PARAMETER NUMBER", "ENTER", ie: 9011 is parameter 11. The table automatically indexes to the next parameter after "ENTER" is pressed each time.

<u>BLOCK #</u>	<u>PROGRAM</u>	<u>COMMENTS</u>	<u>FORMAT</u>
000	Grid location dimension	Grid Position X ⁺	2.4
001	6 digit number	Grid Position Y ⁻	2.4
002	XX.XXXX	Grid Position Z	2.4
003		Grid Position C	2.4
004		Grid Position S ₁	2.4
005		Grid Position S ₂	2.4
006	English software limit dimensions	Limit X ⁺	
007		Limit X ⁻	
008	6 digit number	Limit Y ⁺	
009	XX.XXXX	Limit Y ⁻	
010		Limit Z ⁺	
011		Limit Z ⁻	
012		Limit C ⁺	
013		Limit C ⁻	
014		Limit S ₁ ⁺	
015		Limit S ₁ ⁻	
016		Limit S ₂ ⁺	
017		Limit S ₂ ⁻	
018	XXXX	Rapid Feed (5000 is for 200 IPM) (2500 is for 100 IPM)	
030	XX XX	1st and 2nd Position on feedrate Override Switch	
031	XX XX	3rd and 4th Position on feedrate Override Switch	
032	XX XX	5th and 6th Position on feedrate Override Switch	
033	XX XX	7th and 8th Position on feedrate Override Switch	
034	XX XX	9th and Spare Position on Feedrate Override Switch	

035	XXXX	Power on Feedrate ie: 0200=20 ipm
036	XXXX	Max Feedrate Pulse Rate 5000 = Max

066 Describe grid direction for each axis
6 digit number XX.XXXX
Use the following chart to derive the number

XXXXXX

Bit 0 of this byte is not used for Gridding.
When this bit is "0" Digitizing is disallowed.
In this mode the system will ignore G12FXXXX
command.
When the bit is 1, G12 works normally.

eg: want X+, Y-, Z+ grid
 X+ = 000100)
 Y- = 020200) Add number up
 Z+ = 000400)
 # to insert = 020700 (Grid word)

CHART	
000100	X axis + direction
000200	Y axis + direction
000400	Z axis + direction
001000	C axis + direction
010100	X axis - direction
020200	Y axis - direction
040400	Z axis - direction
101000	C axis - direction
X+	000100
Y-	020200
Z+	000400
	<u>020700</u>

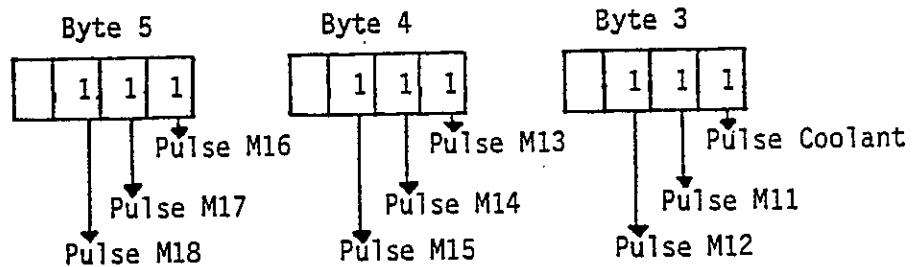
BINARY ANATOMY OF THE GRID WORD

GRID WORD	Direction Bits						Axes select						Not Used
	0 = positive 1 = negative						0 = disable 1 = enable						
	X	X		X	X		X	X		X	X		
	S ₂	S ₁	A	Z	Y	X	S ₂	S ₁	A	Z	Y	X	
Exam 1. 151400	0	0	1	1	0	1	0	0	1	1	0	0	C+Z are enabled in negative direction X+Y are enabled in negative direction X, Y+Z are enabled in positive direction
Exam 2. 030300	0	0	0	0	1	1	0	0	0	0	1	1	
Exam 3. 000700	0	0	0	0	0	0	0	0	0	1	1	1	

BLOCK #	PROGRAM	COMMENTS
067	Metric software limit dimension	Metric limit X+
068	6 digit number	Metric Limit X-
069	XXX.XXX	Metric Limit Y+
070		Metric Limit Y-
071		Metric Limit Z+
072		Metric Limit Z-
073		Metric Limit C+
074		Metric Limit C-
075		Metric Limit S ₁ +
076		Metric Limit S ₁ -
077		Metric Limit S ₂ +
078		Metric Limit S ₂ -
079	Tool change position (inch) 6 digit number with sign +/- XX.XXXX	
080	Grid location dimension	Metric Grid X
081	Metric	Metric Grid Y
082	6 digit number	Metric Grid Z
083	XXX.XXX	Metric Grid C
084		Metric Grid S ₁
085		Metric Grid S ₂
086	Paper tape and RS-232 rates XX.XXXX Use the following chart RS-232 Paper tape 00.000X 00.0xx0 RS-232 X=0=110 baud rate =1=300 baud rate =2=600 baud rate =3=1200 baud rate PAPER TAPE Decitek: XX=22 for 150 cps XX=23 for 300 cps Remex XX=35 for 300 cps EECO XX=06 for 150 cps Spooler XX=07 for 300 cps EECO : XX=20 eg: 00.0233 is Decitek reader at 300 cps and baud rate of 1200	

XXX000

Byte 3,4,5



087 Z tool change position (metric)
6 digit with sign
XXX.XXX

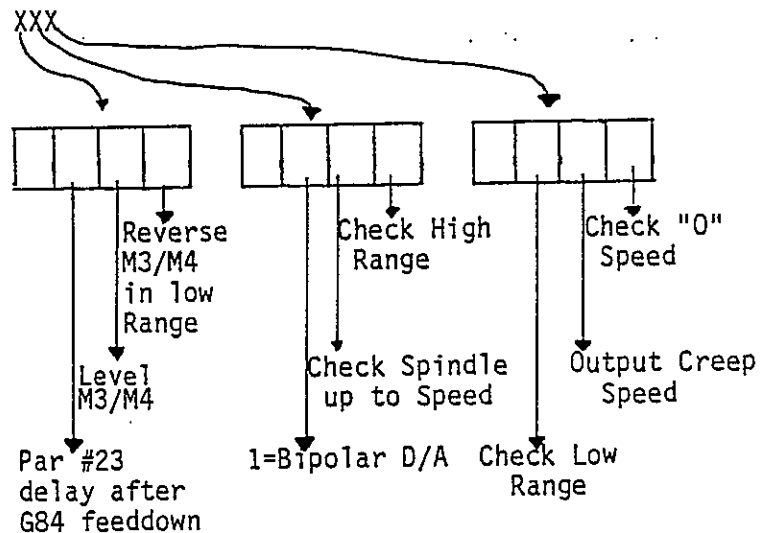
Universal spindle interface: (on systems with option)

088 XXX # steps/RPM High
XXX X steps/RPM Low

089 XXX Max RPM $\div 10$
XXX Shift point RPM $\div 10$

090 XXX Creep Speed in DAC units

1. Bit 0 = 1 will activate spindle override pot.
77.11 and up
76.05 and up
2. Bit 1 = 1 will make M functions go into handshake mode if a P.C. is used with CNC.
3. Bit 2 = 1 will make system not cutter compensate subroutines before cutter compensating the main program. In this mode the programmer must ensure that before calling any subroutine from the main program he uses the M6XX command (see 2-20A).



BLOCK #	PROGRAM	COMMENTS
091	Backlash dimension	Backlash X
092	3 digit number	Backlash Y
093	XXX	Backlash Z
094	Insert the number of 10's	Backlash C
095	of a thousand correction.0xxx	Backlash S ₁
096	maximum number .0256	Backlash S ₂
097	Unidirectional approach	X Unidirectional
098	dimension	Y Unidirectional
099	4 digit number	Z Unidirectional
	XXXX	
	Insert the number of 10's	
	of a thousand correction	
	.XXXX	
	Maximum number .9999	
100	Ball screw error correction dimension	
↓	(see page 15-5)	
483		

I M P O R T A N T

Once the system parameters have been punched in by hand they should be stored on a magnetic cassette labeled "SYSTEM PARAMETER TAPE".

To store the system parameters on tape do a

WRITE TAPE 9 8 ENT

with the correct tape

inserted in the drive. This will store the parameters on the area of tape normally accessed by doing a read or write on Sector 1. So Sector 2 onwards could still be used for parts program storage.

For reading in the system parameter tape whenever the system gives an E07 do

READ TAPE 9 8 ENT

For reading from Paper tape do a

READ TAPE 9 7 ENT.

For dumping to RS-232/TTY do a

WRITE TAPE 9 7 ENT.

PROCEDURE FOR ENTERING BALLSCREW COMP TABLE

FOR X-AXIS

Display Stat #5 tells operator the start comp # table address (usually 100)

Enteries are for every .8192 counts on systems having .0001" = 1 pulse feedback.

1. PREPARATIONS

STEP 1 - Take +ve limit of X and divide by .8192. Take the integer part of the answer and call it N. Call the remainder R.

STEP 2 - Multiply .8192 by N. Call this P1. Now this is the 1st point within the travel limits that the control will allow compensation for.

STEP # - If $R = 0$ (remainder from Step 1), then the error for the first point (P1) can be entered into the comp # displayed in Disp Stat #5. Otherwise if $R \neq 0$ then enter the error for P1 at the comp # displayed in DISP. STATUS #5 plus 1.

2. MEASUREMENTS OF ERROR

NOTE 1 - Always start measuring screw error value from point P1 towards the negative limit for each axis.

Make charts of errors at each point .8192" away from point "P1" up to the negative limit for all axis. Once charts are made the following steps should be followed.

NOTE 2 - At each .8192" interval the control will add to the position the error that it finds in the corresponding entry in the comp table. This necessitates going over the entire chart and noting the most negative error found. Then add to every entry in the table (for that particular axis only) the positive magnitude of that negative error. Now there should be no negative numbers on the table.

Do this for all axis.

NOTE 3 - Backlash compensation plus the largest screw error compensation value for a given axis must never be greater than .0256 inch.

3. ENTERING THE CHARTED ERROR

STEP 1 - Prior to entering one must have calculated point "P1" for each axis along with table entry number (or comp #). Also, screw error charts should be complete and "normalized" to having positive entries only (Note 2).

Going into Display Stat Pg. 9, enter values starting at System Parameters #100 through 300 for all axis.

Now ballscrew comp should be operational.

POINTS TO REMEMBER

1. Make sure there is no backlash or unidirectional approach entries made into the control. They should be zeroed out if previously entered. After completing ballscrew table entry backlash and unidirectional approach entries can be put back in.

EXAMPLE 15.1 TRAVEL LIMITS FOR A MACHINE ARE:

	<u>POSITIVE LIMIT</u>	<u>NEGATIVE LIMITS</u>
X	0	-30.2
Y	11.75	0
Z	0	- 5.6
<u>DISP STAT 5:</u> will show <u>COMP#</u> .		
	X	100
	Y	137
	Z	152

X-AXIS From Step 1: $N=0$
 $0 \div .8192 = 0$ therefore $R = 0$

From Step 2: $P1 = 0$
 $0 \times .8192 = 0$

From Step 3: Enter screw error value for
 $P1$ ($X=0$) into parameter $SP100$

Y-AXIS From Step 1: $N = 14$
since $11.75 \div .8192 = 14.3433$
therefore $R = 3.3433$

From Step 2: $P1 = 11.4688$
since $14 \times .8192 = 11.4688$

From Step 3: Enter screw error value for
 $P1$ ($Y=11.4688$) into parameter $SP138$

Z-AXIS From Step 1: $N = 0$ and $R = 0$

From Step 2: $P1 = 0$

From Step 3: Enter screw error value for $P1$ ($Z=0$)
into parameter $SP152$

So for measuring error, measurements are to be made at the following points.

X-Axis = 0, -.8192, -1.6384, -2.4576, -----, -29.4192

Y-Axis = 11.4688, 10.6496, 9.8304, -----, 0

Z-Axis = 0, -.8192, -1.6384, -----, -4.9152

Where the starting entries for X and Z are made from the comp #
shown in Disp Stat 5 and the one for Y-Axis is made from the Disp
Stat Pg. 5 Comp # +1.

```

484      XXXX }
485      XXXX }      unassigned

```

```

488      XXXX } Logical or these 16 bit
489      XXXX } words to get 16 bit
              } flags

```

```

490      XXXX } Logical or these 16 bit
491      XXXX } words to get 16 bit
                flags.

```

[illegible]

*

15-8

492 XXXX

DMAB

These digits are used to change acc/dec slopes. The system comes standard with the fastest slope possible. The following entries will progressively slow the ramps.

DMAB

0118

1220

2324

0030

3240

4250

3160

6270

5190

0420
corresponds to
60 ipm.
Make number
smaller to
go slower.

493 XXXX X-axis grid speed towards & away from
grid switch

494 XXXX Y-axis grid speed towards & away from
grid switch

495 XXXX Z-axis grid speed towards & away from
grid switch

496 XXXX A-axis grid speed towards & away from
grid switch

497 XXXX B-axis grid switch towards & away from
grid switch

498 XXXX C-axis grid switch towards & away from
grid switch

499 XXXX All axis grid speed towards marker.

500 XXXX Dry run feedrate inch/min. (0....999.9 ipm)
Default=75.0 ipm

501 XXXX Max programmable feedrate. (0....999.9 ipm)
100.0 ipm default

SECTION XVI

DIAGNOSTICS

CENTURION IV SELF-DIAGNOSTIC CAPABILITIES

The Centurion IV has its own self-diagnostic built in. These diagnostics will check the entire ROM and RAM memories, the switches and keypad, and even the handwheel encoder as well as other peripheral equipment. This amounts to about half of the electronics used by the Centurion IV control. What this means to the customer is less down time and smaller service expenses. Since they are self-centered, they are quick and easy to use.

16.1 ROM MEMORY DIAGNOSTICS

The ROM memory is checked every time power is applied to the control. If an E03 is displayed after power is applied to the control, the ROM memory has been checked and is good. A coded row of numbers will appear instead of the E03 if an error occurs.

However, a ROM fault does not necessarily mean that the control will not function or that the control cannot be used by working around the problem area. Whenever an error is displayed, it can be noted, and by depressing any key the error will be cleared and an E03 should be displayed. If a key is not depressed the error will be cleared automatically after about a minute and an E03 displayed.

The ROM test can also be run continuously by holding the "0" key in when power is applied to the control or when the control is Master Cleared. As long as the "0" key is held in, the ROM test will keep repeating itself every few seconds. When this is done the CRT display remains blank unless a ROM fault is detected, at which time the coded row of numbers will appear. These numbers will only flash up on the screen for a couple of seconds, however, unless the error occurs consistently. When the "0" key is released an E03 should be displayed.

The numbers that may appear during a ROM fault and what they mean are as follows:

- 0 = good ROM location
- 1 = ROM failure in an XXd EPROM where XX is the EPROM name (this name is only alpha-numeric).
- 2 = ROM failure in an XXU EPROM (each pair of EPROM have the same name except for the "U" and "d" postscripts).
- 3 = ROM failure in both the "U" and "d" EPROM's having the same name.

EXAMPLE 16.1.1 ROM TEST FAILURE

ROM Address	0 u & d	8 u & d	10 u & d	18 u & d
	0 0 0 0 0 0	0 1 0 0 0 0	0 0 0 0 0 0	0 0 0 0 0

In this example, since a "1" occurred in the 9th digit it means there was a ROM failure in EPROM 8d.

16.2 RAM MEMORY TEST

By depressing the "1" key during power up or when Master Clearing the control the RAM test will begin to run. This is the memory which contains the parts programs and the more memory available the longer it takes this test to run (10 to 15 minutes is typical).

While this test is running, the words "RAM Test" will flash on and off on the CRT screen. When the test is completed, the words "RAM Test" remain but an E07 is also displayed. At this time the RAM memory has been erased and it may be necessary to load the System parameter tape.

If a RAM failure occurs there will be three four digit numbers displayed. The four digit number on the left indicates the faulty address, the four digit number in the center indicates what was written into the RAM at this address, and the four digit number on the right indicates what was read back from the address. Normally, the number written into RAM will agree with the number read from ROM. By noting the difference between these two numbers and the address it can be determined which RAM chip is faulty.

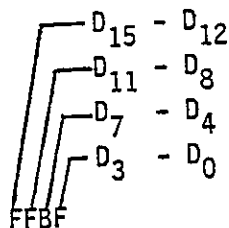
EXAMPLE 16.1.2 RAM ERROR

CRT Display will say:

RAM TEST

3E00

FFFF



FFBF

ADDRESS

DATA WRITTEN

DATA READ

The address determines which row of RAM is faulty and the difference between data read and written determines which chip in that row is faulty. In this it would be the chip responsible for signals D₇ - D₄.

16.3 DISPLAY STATUS PAGE 5 (Keypad Test)

This is the I/O page and gives a representation of nearly all inputs to the control (see Display Status feature). The keypad has a hex decimal coded decimal test code for each key. Table 1 shows the test code for each key. The test code for a specific key should be displayed, in parallel channels 00 (upper character) and 0 (lower character) on the I/O page, (See Section 2.3.5), whenever the key is depressed.

TABLE 1

KEY	0	1	2	3	4	5	6	7	8	9	±	.	CE	ENT
TEST CODE	0	1	2	3	4	5	6	7	8	9	B	A	1F	1E

KEY	F	S	Z	Y	X	R	G	M	MDI	DISP OFFSET	EDIT ADD	PROG DISP	CLR PRG	
TEST CODE	24	26	23	22	21	20	25	27	14	11	13	16	1A	

KEY	DEL	EOB	READ TAPE	WRITE TAPE	DISP STAT	GRID	JOG	RUN	DRY RUN	EN HDW	RAPID JOG			
TEST CODE	10	12	18	19	10	1C	18	17	0E	00	0C			

16.4 ERASE MAGNETIC TAPE AND FORMATE

To erase a magnetic tape turn power on with the Write Tape key depressed. The control is now waiting for a cassette tape to be inserted into the tape reader. If the tape switch is in the "Record" position it will be erased. When the erase sequence is finished, removal of the tape from the reader will terminate the erase sequence and an E03 will be displayed. If there are several tapes to be erased, holding the Write Tape key in when removing a tape will keep the control in the erase sequence. If the tape is brand new you must manually wind it past 2 sets of end of tape holes. If this is not done the tape will run off the end if erasing is attempted. Then using the write tape procedure (5-11) write sections 01 thru 09 in order or write until an E14 appears. The E14 error indicates that the maximum number of tape sectors has been reached.

SECTION XVII

PROGRAMMING PROBLEMS

The following examples give a variety of programming problems and explain how these problems are solved using the control. Each example program includes the precise start and end points for the contour moves. Some of the examples will give an alternative program which will show the advantage of approximating start and end points rather than having to calculate these points.

In the first 8 examples, the dotted line around the part outline indicates the path of the center of the tool as programmed around the part. The capital letters shown on the diagrams indicate the sequence of the cutter as programmed around the part. and the small letters indicate the sequence of the actual tool path around the part.

17.1 ANGLES TANGENT TO ARCS

<u>POS.</u>	<u>BLOCK #</u>	<u>BLOCK ENTRY INFO. - PROGRAM</u>
A	1001	G101, G0, M7, F10, X-1, Y-1
A	1002	G300, P.1
A	1003	G308, P-.15
B	1004	S1500, M3, G86, G41, X0, Y-1
C	1005	G1, Y3.5
D	1006	X1.5
E	1007	G3, R1, I2.5, J3.5, X3.207, Y2.7929
F	1008	X5.2071, Y4.7929
G	1009	G2, R1, I5.9142, J4.0858, X6.9142, Y4.0858
H	1010	Y0
O	1011	X0
I	1012	G99, G80, G0, G40, Y.5
O	1013	M5, M9, G100, G98

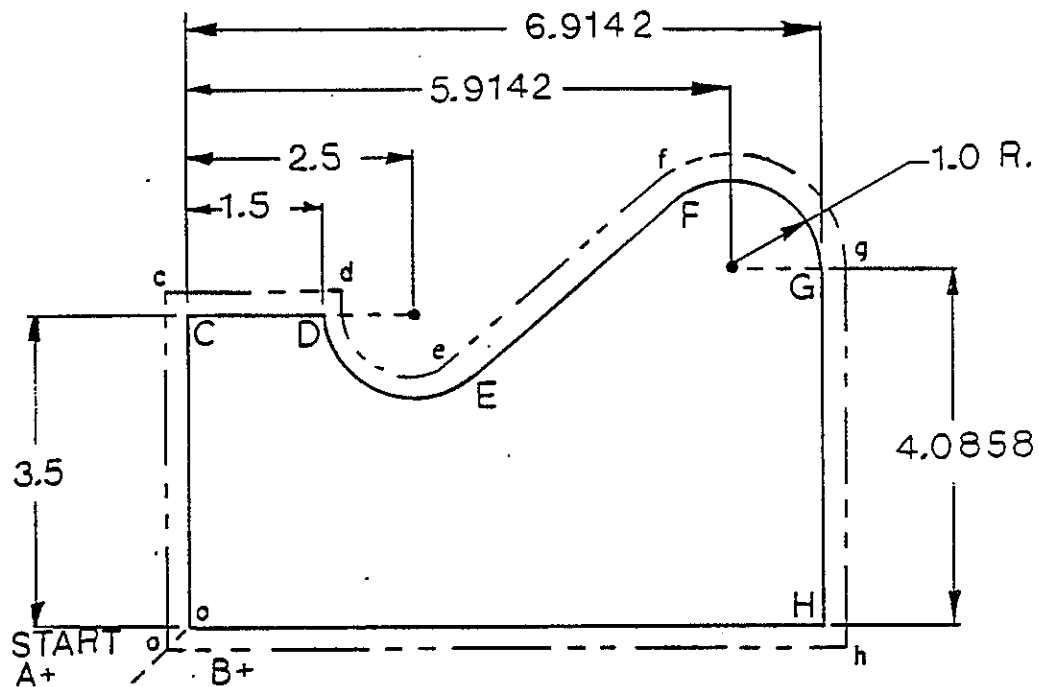


Figure 17.1 ANGLES TANGENT TO ARCS

The set-up procedure for this part would be:

1. Locate and set the X and Y Floating Zero in the lower left corner of the piece.
2. Enter the tool radius into the appropriate tool offset location (determined by tool # to be used).
3. Activate this tool with the G1XX code.
4. DRY RUN the program to see if it is correct and complete. For a more detailed set-up procedure, see Section VII 7.3.

1001 Activate Tool #1, tool change

1002 Set R-Plane

1003 Set Z depth

1004 Start Spindle

- 1005 Turns ON the feedrate (G1) programmed in block 1, and moves at this feedrate to position "C".
- 1006 to Causes the tool to move around the part piece.
- 1011
- 1012 Z rapids to the R-Plane and the milling cycle is cancelled. It then turns OFF the cutter compensation (G40) before making a rapid move to position "I". This move is necessary in order to define the corner of the workpiece to the control so the final cutter compensated move is complete. It would suffice to make this move just .001 inch long.
- 1013 Shuts OFF the spindle (M5) and coolant (M9), and causes a rapid move back to Floating Zero (G98).

17.2 TANGENT ARCS

POS.	BLOCK #	BLOCK ENTRY INFO-PROGRAM
A	2001	G102, F10, G0, M7, X.2, Y.2
A	2002	G300, P.1
A	2003	G308, P-1.2
B	2004	S1400, M3, G86, G42, X0, Y.2
C	2005	G1, Y-1.5
D	2006	G3, R1, X1, Y-1.5, X1.7071, Y-2.2071
E	2007	G2, R1.4143, X2.7071, Y-3.2071, X3.7071, Y-2.2071
F	2008	G3, R1, X4.4142, Y-1.5, X5.4142, Y-1.5
G	2009	Y0
O	2010	X0
H	2011	G99, G80, G40, G0, X0, Y-.1
	2012	M5, M9, G100, G98

ALTERNATE PROGRAM

	2001	G102, M06
A	2002	F10, G0, M7, X.2, Y.2
A	2003	G300, P.1
A	2004	G308, P-1.2
A	2005	S1200, M3
A	2006	G86
B	2007	G42, X0, Y.2
C	2008	G1, Y-1
D	2009	G3, R1, I1, J-1.5, X1., Y-2.5
E	2010	G2, R1.4143, I2.7071, J-3.2071, X2.7071, Y-1.7929
F	2011	G3, R1, I4.4142, J-1.5, X5.4142, Y-1.5
G	2012	Y0
O	2013	X0
H	2014	G99, G80, G40, G0, X0, Y-.1
	2015	M5, M9, G100, G98

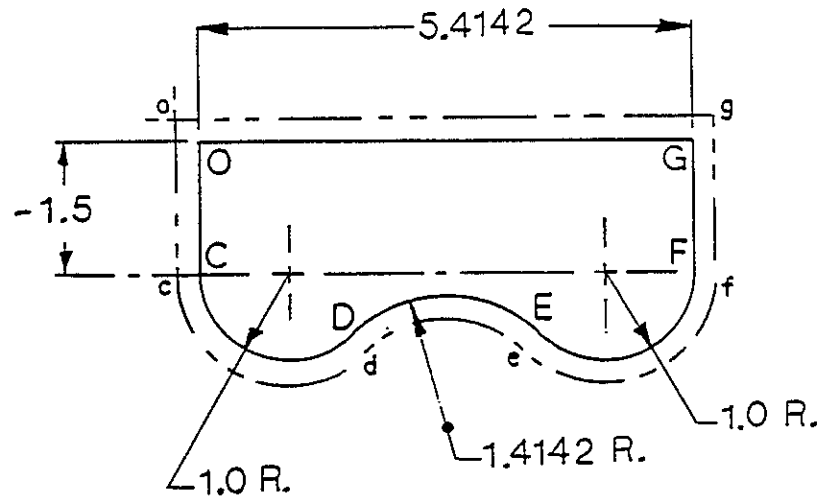


Figure 17.2 TANGENT ARCS

The set-up procedure for this program is the same as in 17.1, but this time the Floating Zero is set in the upper left hand corner of the part piece.

- 2001 Activates Tool #2,
Programs a rapid move to position "A". Also establishes a feedrate of 10 IPM and turn ON the coolant.
- 2002 Set R-Plane
- 2003 Set Z depth
- 2004 Start Spindle
Enable Mill Cycle
Turns ON right cutter compensation (G42) and brings the tool into position to begin cutting. Note the difference of this starting position to that of 17.1. In this case, the distance from "B" to "O" will determine how far away from the part the tool will come into position and this distance will not be determined by cutter size, but by the amount of stock to be removed.
- 2005 to Accomplish the cut to be made
- 2010
- 2011 Causes a rapid move to position "H". It also shuts OFF the cutter compensation
- 2012 Shuts OFF the spindle and coolant, and returns all axes to Floating Zero

Alternate program 17.2 will work in precisely the same way as the program just described. The only differences are in the dimensions programmed in blocks 8, 9 and 10. These are convenient approximations which require no trigonometric calculations, but which will allow the computer to determine the correct intersection points of the arcs and angles described.

17.3 TANGENT ARCS AND ANGLES TANGENT TO ARCS

POS.	BLOCK #	BLOCK ENTRY INFO - PROGRAM
A	3001	G103, F10, G0, M7, X0, Y.2
A	3002	G300, P.1
A	3003	G308, P-.75
O	3004	S1000, M3, G86, G41, X0, Y0
B	3005	G1, X1
C	3006	G2, R1, I2, J0, X1.2929, Y.7071
D	3007	X3.2929, Y2.7071
E	3008	R1, I4, J2, X5, Y2
F	3009	G3, R2, I7, J2, X9, Y2
G	3010	Y4.5
H	3011	X0
O	3012	Y0
I	3013	G99, G80. G40, G0, X.2, Y0
O	3014	M5, M9, G100, G98

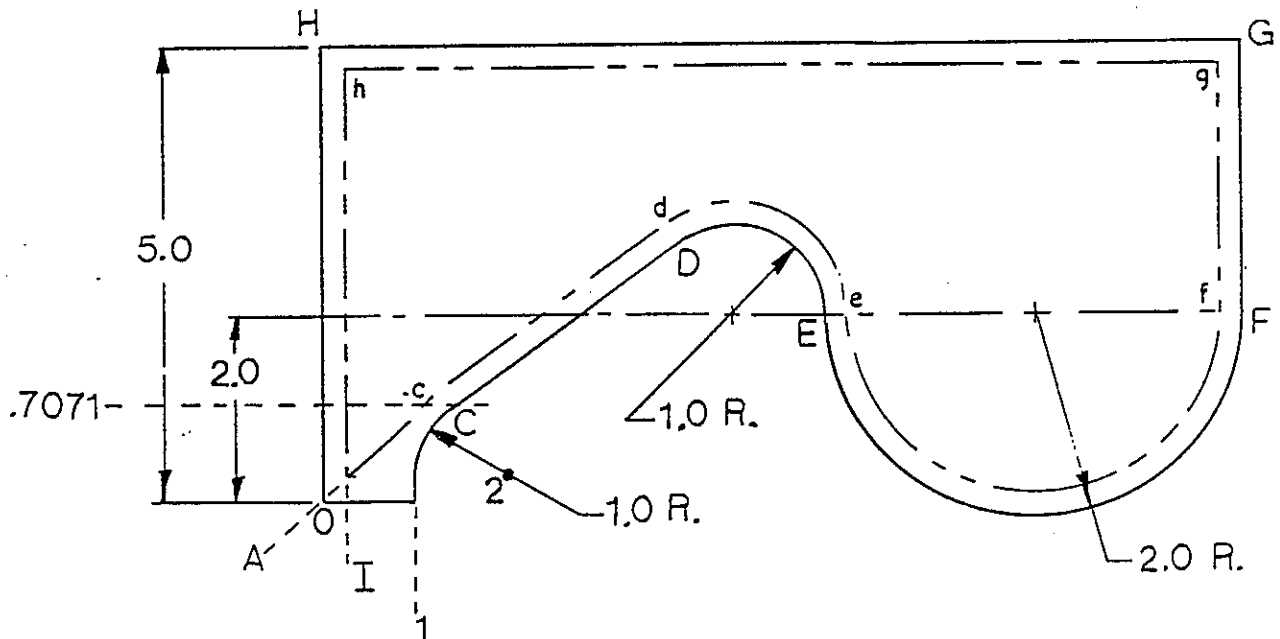


Figure 17.3 TANGENT ARCS AND ANGLES TANGENT TO ARCS

The set-up procedure for this program is the same here as in the previous examples.

- 3001 Activates Tool #3,
Programs a rapid move to position "A". Also establishes a feedrate of 10 IPM and turns ON the coolant.
- 3002 Sets the R-Plane
- 3003 Sets Z depth
- 3004 Starts the Spindle
Enables the Milling Cycle
Due to the direction of cut, which is in a counterclockwise direction, this block establishes an inside cut. It also allows the tool to be brought down to cutter depth at position "O" which is directly in the corner of the part. Positions "A" and "I" could have been chosen to be anywhere along their respective edges to cause the tool to start and finish its cut at position "O".
- 3005 to Takes the tool around the inside of the part at 10 IPM
- 3012
- 3013 Causes a program stop to raise tool and a rapid move to position "I". It also shuts OFF the cutter compensation.
- 3014 Makes a rapid move to the Floating Zero point and shuts OFF the spindle and coolant. This position could be used to replace the tool with a finish cutter or to load new parts.

17.4 NON-TANGENT ARCS

POS.	BLOCK #	BLOCK ENTRY INFO - PROGRAM
	4001	G104,
A	4002	F10, G0, M7, X.001, Y0
A	4003	G300, P.1
A	4004	G308, P-1.0
A	4005	S1200, M3
A	4006	G86
O	4007	G41, X0, Y0
B	4008	G1, Y-2
C	4009	G2, R1, I0, J-3, X.7071, Y-3.7071
D	4010	G3, R2, I2.1213, J-2.2929, X3.5355, Y3.7071
E	4011	G2, R1, I4.2426, J-3, X4.2426, Y-2
F	4012	Y0
O	4013	X0
G	4014	G99, G80, G40, G0, X0, Y-.001
O	4015	M5, M9, G100, G98

ALTERNATE PROGRAM

	4001	G104,
A	4002	F10, G0, M7, X.001, Y0
A	4003	G300, P.1
A	4004	G308, P-1.0
A	4005	S1200, M3
A	4006	G86
O	4007	G41, X0, Y0
B	4008	G1, Y-2
C	4009	G2, R1, I0, J-3, X1, Y-3
D	4010	G3, R2, I2.1213, J-2.2929, X2.1213, Y-4.2929
E	4011	G2, R1, I4.2426, J-3, X4.2426, Y-2
F	4012	Y0
O	4013	X0
G	4014	G99, G80, G40, G0, X0, Y-.001
O	4015	M5, M9, G98

This set-up procedure is the same as in Example 17.1

4001	Activates Tool #4, Tool Change
4002	Establishes the start point of the cut at position "0"
4003	Sets the R-Plane
4004	Sets Z depth
4005	Starts the Spindle
4006	Enables the Milling Cycle
4007	Establishes the start point of the cut at position "0"
4008	
to	Takes the tool around the inside of the part at 10 IPM
4013	
4014	Provides the cutter compensation feature, the necessary information to accurately complete the cut to the precise starting point of the cut. Rapids Z to the R-Plane and cancels the Milling cycle.
4015	Rapids to Floating Zero point and turns OFF Spindle and Coolant.

Again this alternate program 17.4 is the same as the original program, but does not require trigonometric calculation of points "C and D". The end points in blocks 5 and 6 are picked at the 90° points and the computer calculates the correct arc intersection points.

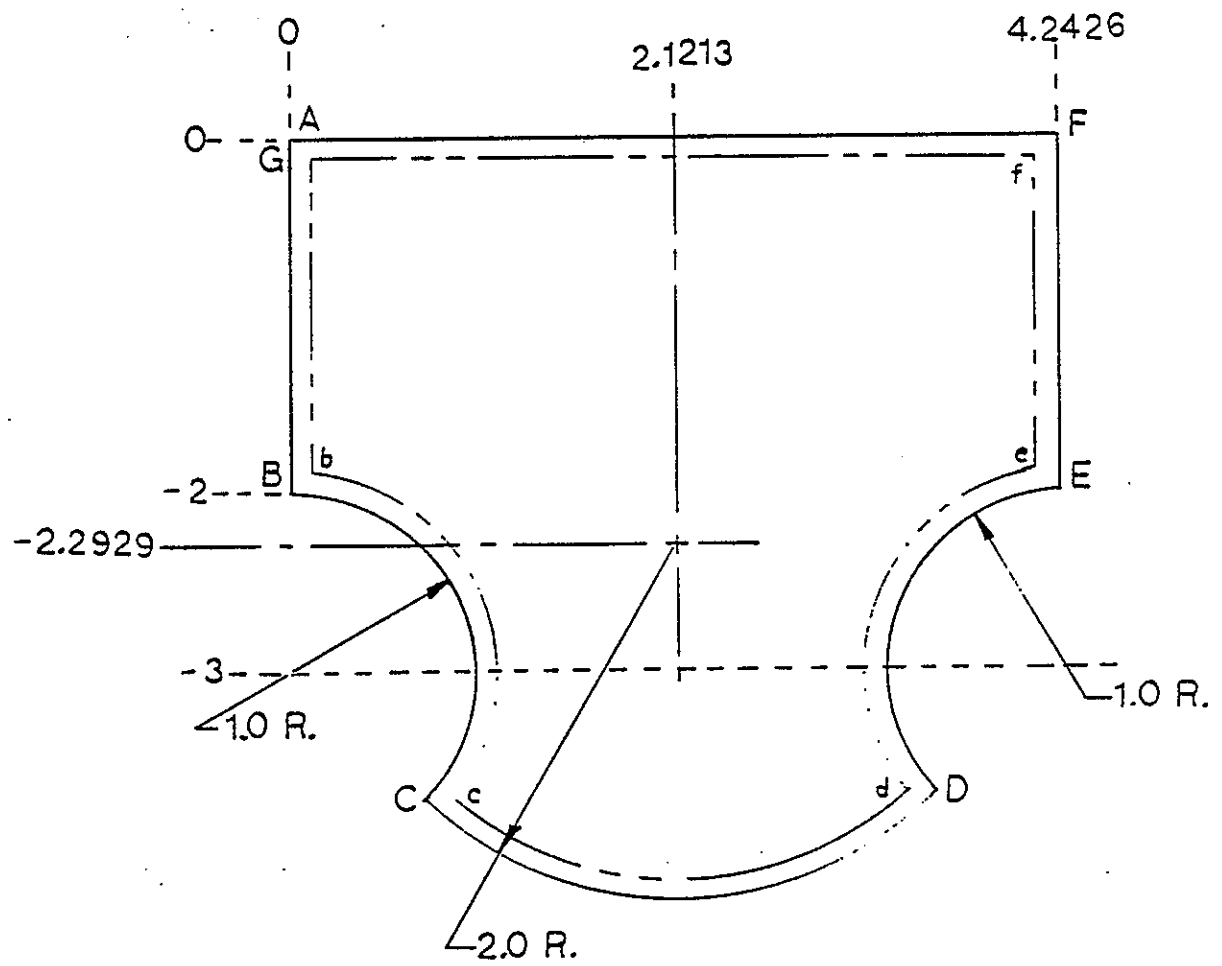


Figure 17.4 NON-TANGENT ARCS

17.5 NON-TANGENT ARCS

POS.	BLOCK #	BLOCK ENTRY INFO-PROGRAM
A	5001	G105, F10,G0,M7,X.001,Y0
A	5002	G300, P.1
A	5003	G308, P-.4
O	5004	S1500, M3,G86,G42,X0,Y0
B	5005	G1, Y2
C	5006	G2, R2, X2, Y2, X3.4142, Y3.4142
D	5007	R2, X4.8284, Y2, X6.8284, Y2
E	5008	Y0
O	5009	X0
F	5010	G99, G80, G40, G0, M5, M9, X0, Y.001
O	5011	G100, G98

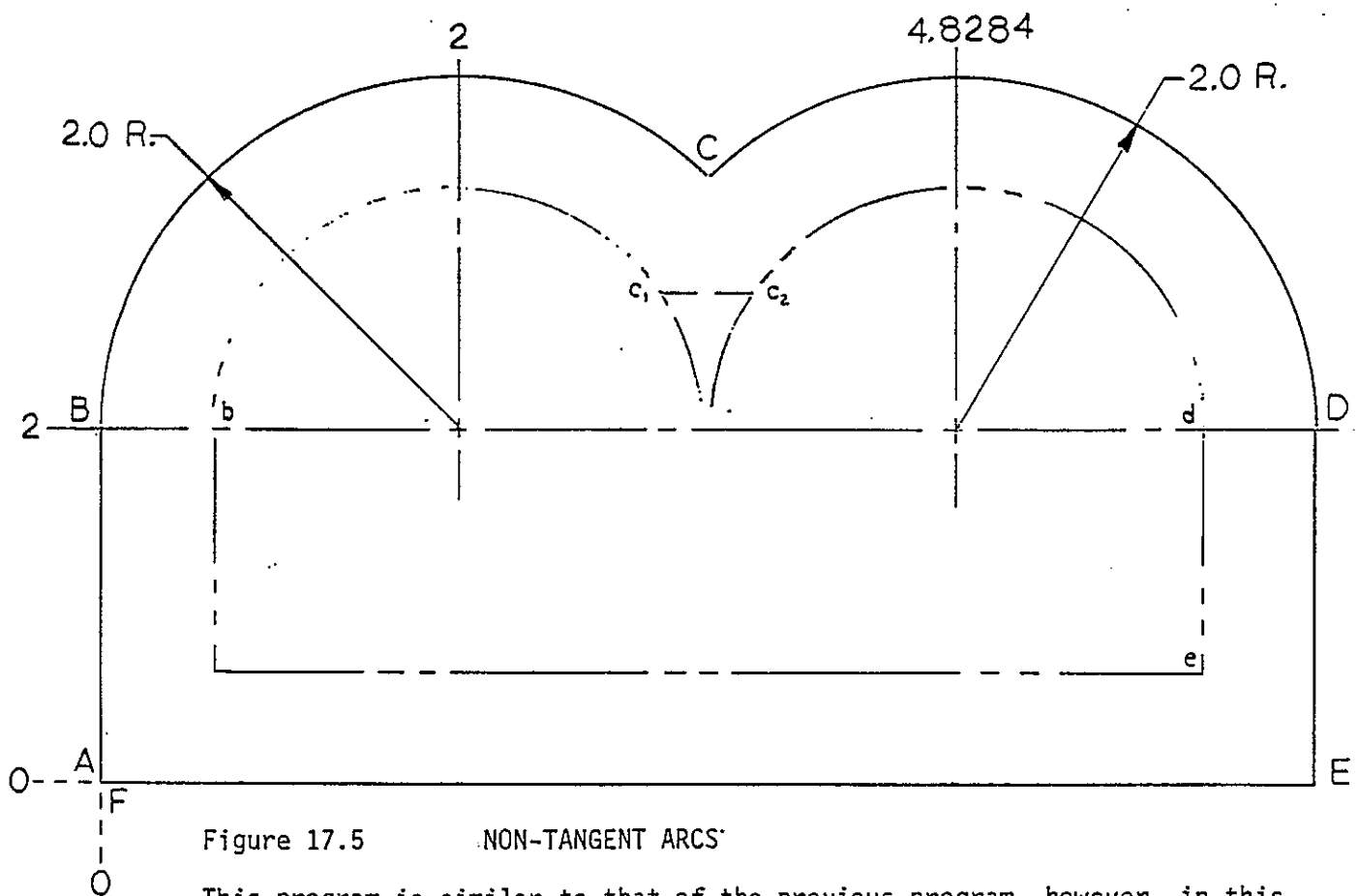


Figure 17.5 NON-TANGENT ARCS

This program is similar to that of the previous program, however, in this example, if the tool size is greater than 1.25 inches, there will be no intersection point at position "C". Therefore, the program would have to be altered for a large tool by inserting a block between blocks 10 and 11 of the program, which would simply command a move in X to 3.4143. This accomplishes an undetectable move of .0001 inch, but does force an intersection point between the 2 arcs. Therefore, the cutter path would now be from C₁ to C₂ and the part could now be accomplished with any size tool.

17.6 NON-SQUARE CORNER START POSITION

<u>POS.</u>	<u>BLOCK #</u>	<u>BLOCK ENTRY INFO-PROGRAM</u>
	6001	G106,
	6002	G0, X.999, Y2.997
A	6003	G300, P.1
A	6004	G308, P-.25
A	6005	S1500, M3
A	6006	G86
A	6007	G42, M7, X1, Y3
B	6008	F10, X0
C	6009	G3, R1.5, I-1.2689, J2.2, X-2.5378, Y3
D	6010	X-4
E	6011	X-6.1, Y.5
F	6012	Y0
O	6013	X0
A	6014	X1,Y3
A	6015	G99, G80, G0, G40,M5,M9,X.999,Y3
	6016	G100, Y4

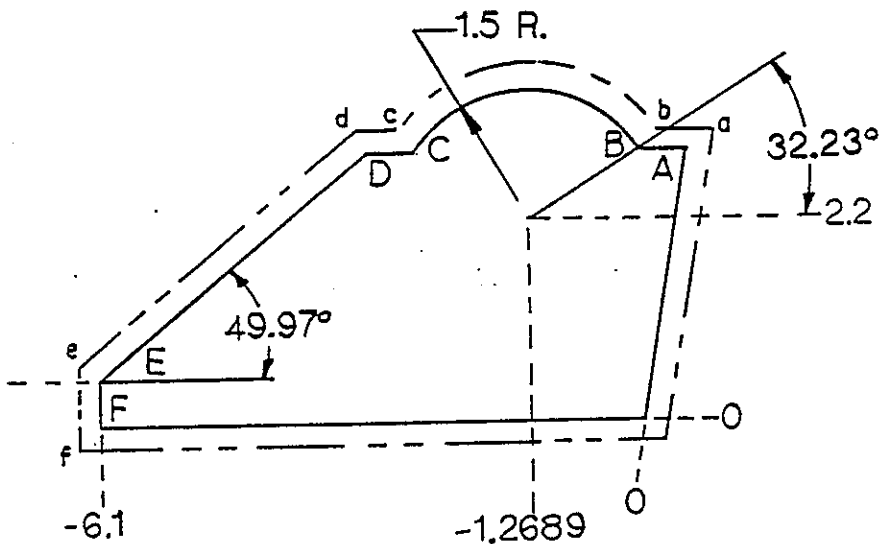


Figure 17.6 NON-SQUARE CORNER START POSITION

In this example, a start point is picked on the slope of the line between points "O" and "A". This insures that the cutter will begin on the correct cutter path and will end at this same point to insure the full pass around the part no matter what size cutter is used.

17.7 CONCENTRIC ARCS HAVING EQUAL RADII

POS.	BLOCK #	BLOCK ENTRY INFO-PROGRAM
	7001	G107,
	7002	G300, P.1
A	7003	F10, G0, X1.25, Y-.2
B	7004	G99, M7, G42, X1.25, Y0
B	7005	S2000,M3,G1,Z-.2
C	7006	G3, R1.25, I0, J0, X0, Y1.25
C	7007	Y1.2501, Z-.15
D	7008	R1.25, I0, J0, X0, Y-1.25
D	7009	Y-1.2499, Z-.2
B	7010	R1.25, I0, J0, X1.25,Y.2
E	7011	G99, G80, G0, G40, Y.2
O	7012	M5, M9, G100, G98

Tool size = .375

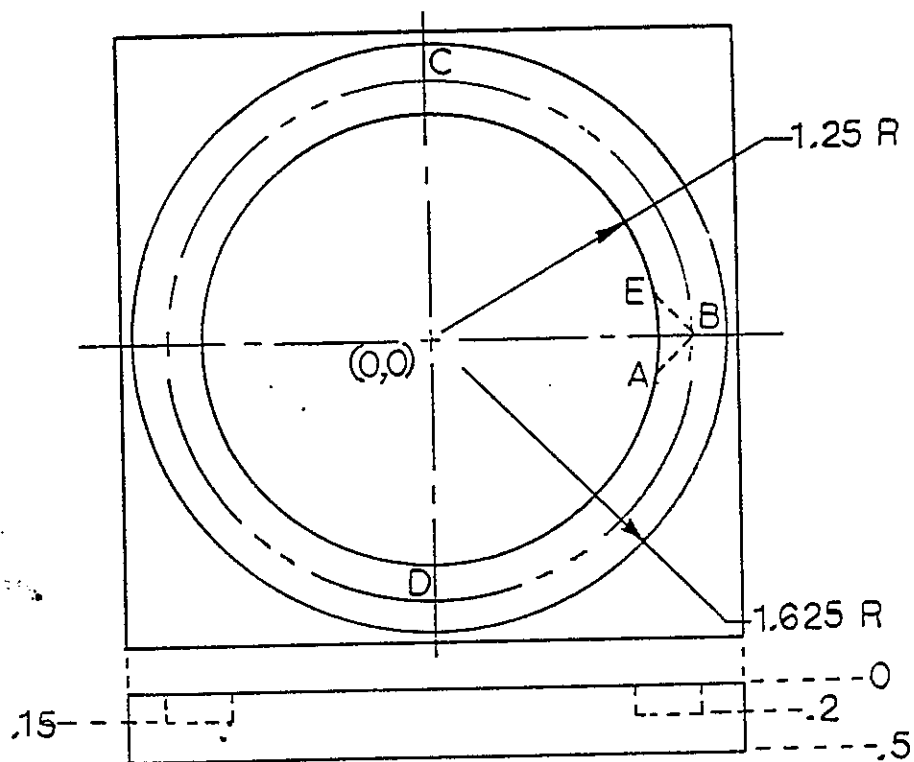


Figure 17.7

CONCENTRIC ARCS HAVING EQUAL RADII

EXAMPLE

USING A TOOL SMALLER THAN .325 INCH

<u>POS.</u>	<u>BLOCK #</u>	<u>BLOCK ENTRY INFO-PROGRAM</u>
	7001	G107,F10,G99,G0,X.2,Y-1.625
	7002	G42, X0, Y-1.625
	7003	G1, Z-.2
	7004	M3, M7, Y-1.25
	7005	G3, R1.25, I0, J0, X0, Y1.25
	7006	Y1.625
	7007	G2, R1.625, I0, J0, X0, Y-1.625
	7008	Y-1.25
	7009	G99, G40, G0, X.2
	7010	X-.2, Y-1.625
	7011	G41, X0, Y-1.625
	7012	G1, Z-.15
	7013	Y-1.25
	7014	R1.25,I0,J0,X0,Y1.25
	7015	Y1.625
	7016	G3, R1.625, I0, J0, X0, Y-1.625
	7017	Y-1.25
	7018	G99, G0, G40, X-.2
	7019	M5, M9, G98, G100

In this example, a circular slot is to be cut having 2 depths. If a tool is used that is the same size as the slot, the slot can be accomplished by a simple circular program. However, the circular program must be stopped at the points where the depth changes, to allow a Z -axis move, and then restarted. This cannot be done without making a Y-axis move in between the circular moves and should only be attempted at the 90°, 180°, 270° and 360° quadrant points of a circle.

7001 Activates Tool #7, Tool Change
7002 Sets R-Plane
7003 Sets Feedrate at 10 IPM, and the tool to rapid to position "A"
7004 Z rapids to the R-Plane. Right Cutter Compensation is turned ON and a rapid move is made to position "B", and coolant is turned ON.
7005 Spindle is turned ON. Feedrate is turned ON, Z-axis is brought down to cutting depth.
7006 A 90° arc to position "C"
7007 Z-axis is brought up to the new depth and a .0001 inch move is made in Y to meet the requirement stated above and to define the cutter compensated path.
7008 180° arc is programmed to position "D"
7009 Z-axis is brought down to the lower cutting depth and a Y-axis move made to meet the concentric arc requirement, and to define the cutter compensated path.
7010 a 90° arc is commanded back to position "B"
7011 A rapid move is made to position "E" after the tool is retracted
7012 Cutter compensation, spindle, and coolant are turned OFF, and a rapid move made back to Floating Zero.

17.8 FIREHYDRANT/MIRROR IMAGE

POS.	BLOCK #	BLOCK ENTRY INFO-PROGRAM
	8001	G108, M06
	8002	G300, P.1
	8003	G308, P-.5
	8004	S1500, M3
	8005	M7, G0, X0, Y0
	8006	G801, G61, G801, G62, G801
	8007	G60, G62, G801, G60,
	8008	G65, G801, G61, G801, G62, G801, G60, G65, G62, G801, G60, G100, G98
	9001	G701
A	9002	G0, X1.001, Y1
A	9003	G86
B	9004	G41, X1, Y1
C	9005	G1, F10, Y2.5
D	9006	G2, R1, I2, J2.5, X3, Y2.5
E	9007	G3, R.25, I3.25, J2.5, X3.25, Y2.25
F	9008	G2, R.25, I3.25, J2, X3.25, Y1.75
G	9009	G3, R.25, I3.25, J1.5, X3, Y1.5
H	9010	Y1
B	9011	X1
J	9012	G99, G80, G40, G0, Y1.001
O	9013	X0, Y0
	9014	G701

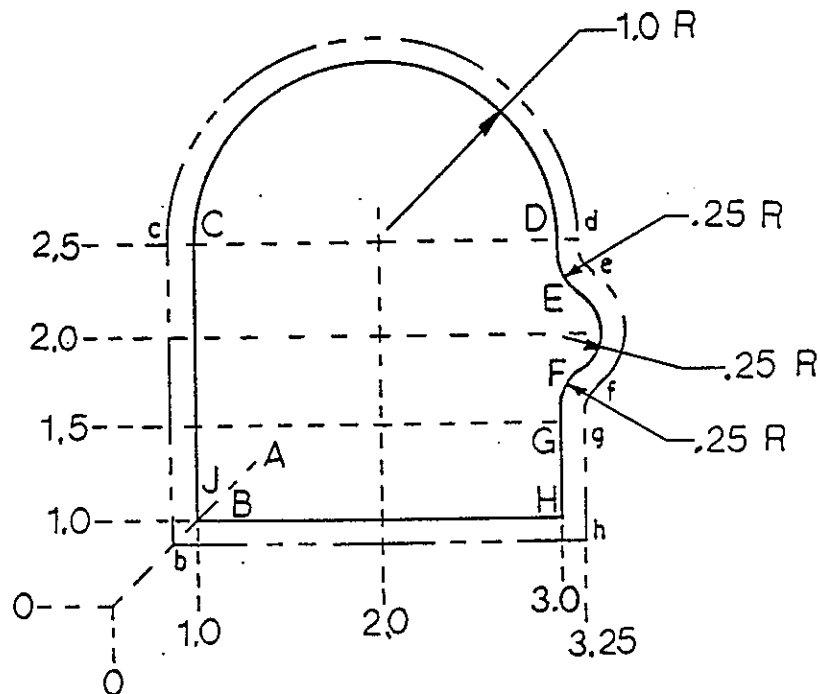


Figure 17.8 (a) FIREHYDRANT/MIRROR IMAGE

In this example, the "firehydrant" routine is placed in a subroutine. This allows the operator to write a short program to turn the X and Y mirror images ON and OFF, and repeat the firehydrant routine at will by simply calling up subroutine #1 (G801).

```

8001 Activate Tool #8, Tool Change (this insures the subroutine is cutter
8002 Set R-plane                                     compensated for tool #8).
8003 Set Z depth
8004 Spindle start
8005 Coolant is turned ON and a rapid move made to Floating Zero
8006 Subroutine #1 is called up(G801) which causes this routine to
      be accomplished in the upper right hand quadrant. Then the X
      mirror image is turned ON (G61) and the routine is repeated in
      the upper left hand quadrant. Now the Y mirror image is turned
      on(G62) and the routine repeated again in the lower left hand
      quadrant.
8007 Turns the mirror images OFF (G60) and then turns ON the Y mirror
      image only. The routine is repeated once again, but now in the
      lower right hand quadrant.

```

Note that blocks 6 and 7 could have been entered in a single block, since no axes dimensions are required in either block.

Also note that the G60 in block 8007 is only necessary to provide a buffer between the G801 command the M-Function commands. If the G60 command were left out of this block, the subroutine would be looped on 5 times instead of turning the spindle OFF.

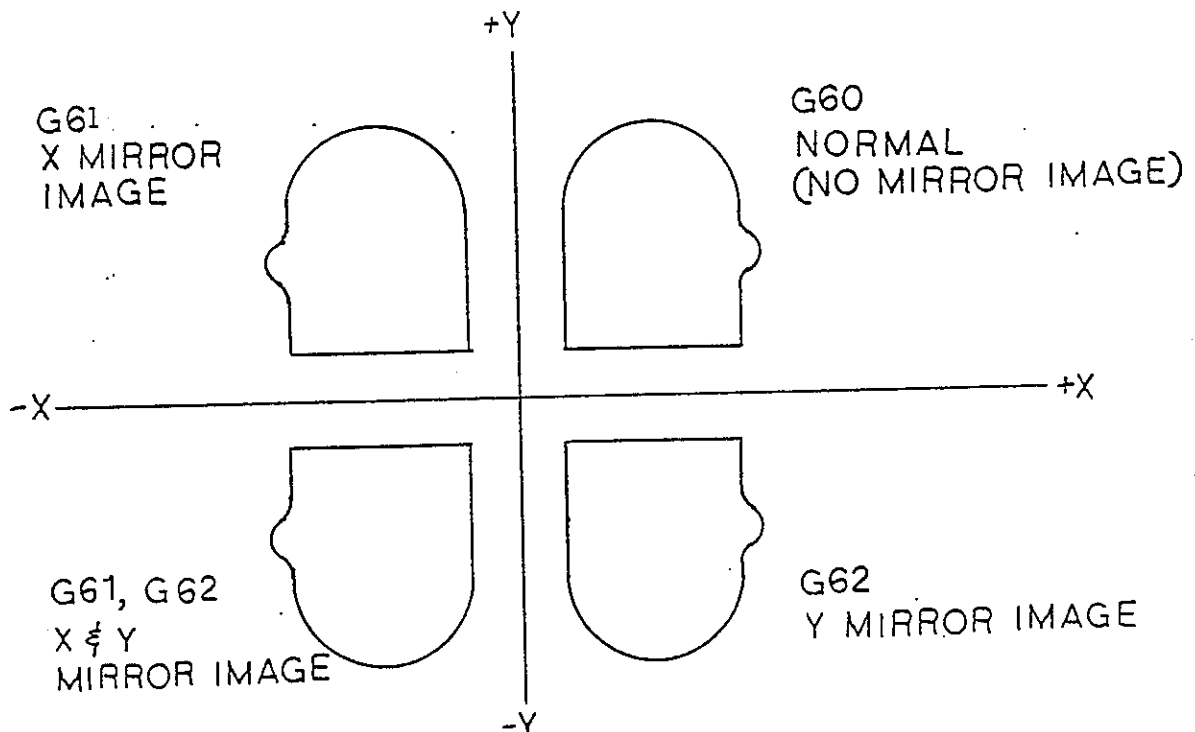


FIGURE 17.8(b)

MIRROR IMAGE

8008 For the explanation of this line reference Figure 17.8 (c). Axis interchange is turned on (G65) which causes the figure in the upper right hand quadrant to be performed by subroutine #1 (G801). This figure is then mirror imaged in X (upper left quadrant), X and Y (lower left quadrant), and Y (lower right quadrant) to complete the program.

NOTE: G65 and G61 cause the original part (figure 17.8a) to be rotated 90° ccw; G61 and G62 cause the part to be rotated 180° ccw; and G65 and G62 cause the part to be rotated 270° ccw.

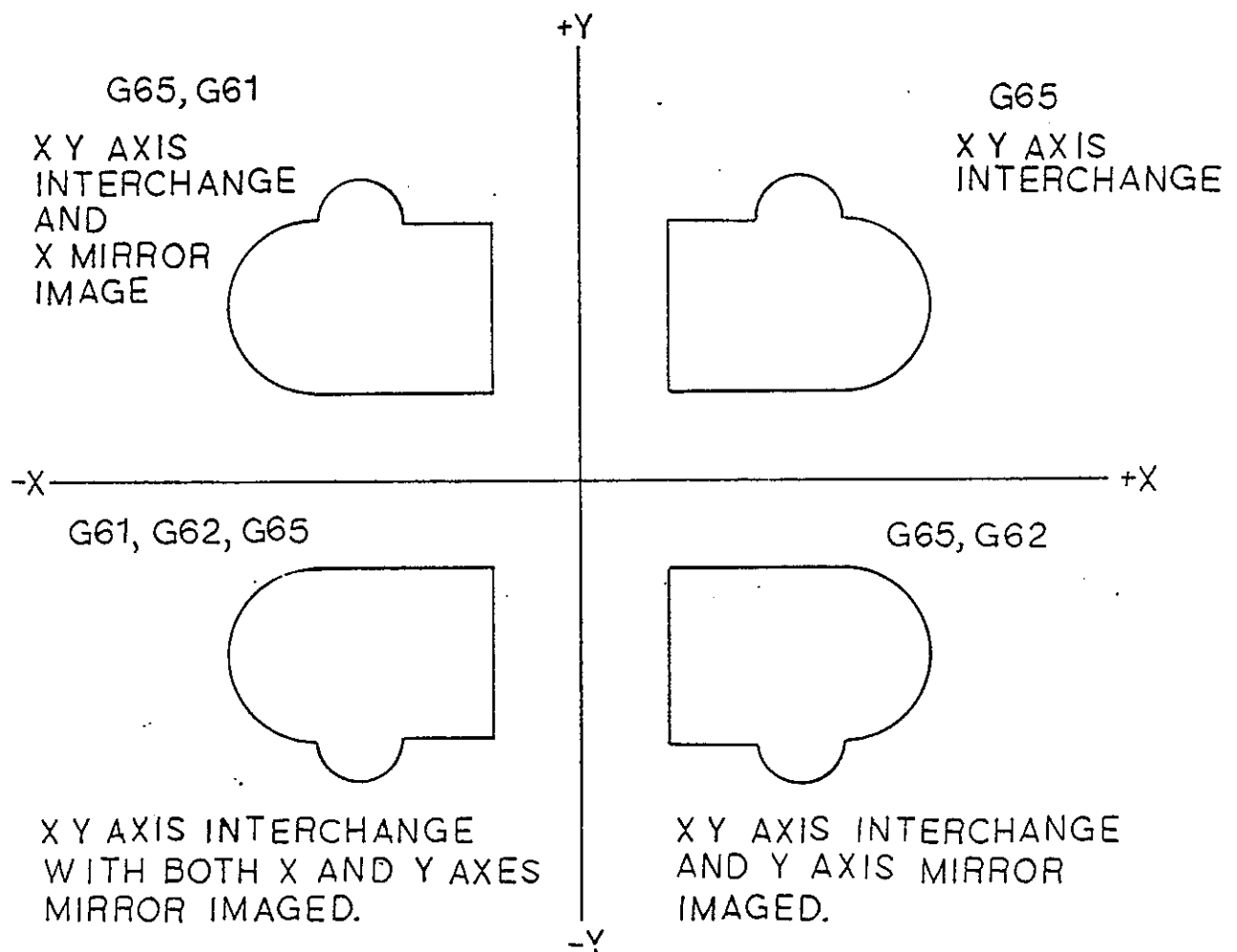


FIG. 17.8(c)

17.9 HOLE DRILLING AND TAPPING PATTERN

BLOCK #	BLOCK ENTRY INFO-PROGRAM
1001	G109, M06
1002	G300, P.1
1003	G99, F10, G0, X2, Y3.5
1004	G308, P-.75
1005	S200, M3, M8, G81, G802, X3, Y4
1006	G81, G802, G98
1007	G119, M6, S200, M3, M8
1008	G0, X2, Y3.5
1009	G308, P-.5
1010	G323, P.5
1011	G84, G324, P.02
1012	G802, X3., Y4.
1013	G84, G802, M5, M9, G100, G98
1014	G323, P0
1015	G324, P0
9015	G702
9016	G301, P2
9017	G302, P5
9018	G305, P1
9019	G304, P0
9020	G27, G80, G99
9021	G702

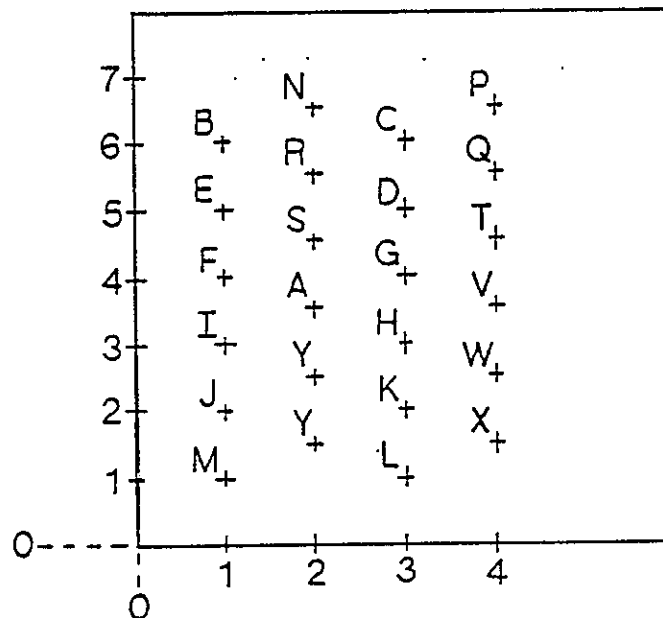


Figure 17.9

HOLE DRILLING AND TAPPING PATTERN

Here again, we use a subroutine to make the program shorter and simpler.

```

1001 Activates Tool #9, Tool Change
1003 Commands a rapid move to position "A" which is the center of the
      1st 12 holes after sending Z to the R-Plane (G99).
1005 Calls up subroutine #2 (G802) which does holes "B" through "M" and
      makes a move to position "G" which is the center of the 2nd set of
      holes
1006 Enable Drill Cycle
      Again calls up the subroutine which does holes "N" through "Z" and
      then a move to Floating Zero for a tool change (M6)
1007 The spindle and coolant are turned ON, after the tool change
1008 Rapid to center of holes
1009 Z-depth for the tapping cycle
1010 Dwell at bottom of tap
1011 Enable tapping cycle and sets feedrate for canned cycle at 20ipm
      (G324,P.02).
1012 to Tap all 24 holes by using the subroutine
1013 Stops the spindle, turns OFF coolant, rapids back to Floating Zero

9015 Labels the subroutine #2 (G702)
9016 Establishes the G301 and G302 parameters which are equal to the length
      to of any row of holes and the distance between the 1st and 3rd or 2nd
9019 and 4th rows, G304 is used to Zero P4. Parameter G305 establishes
      the Y incremental, which is equal to the distance between holes.
9020 Selects the rectangular pocket clearing routine (G27) which will be
      used for a symmetrical hole drilling pattern in this case.
9021 End of subroutine #2

```

17.10 BOLT HOLE PATTERN

<u>POS.</u>	<u>BLOCK #</u>	<u>BLOCK ENTRY INFO-PROGRAM</u>
	2001	G110, M06
	2002	G98, F10, G0, M8
	2003	G300, P.1
	2004	G308, P-.5
	2005	G323, P1
	2006	S100, M3
	2007	G82
	2008	G303, P1.5
	2009	G310, P4.5
	2010	G3, G803, L2, G80, M5, M9, G303, P0
	2011	G100, G98, G323, P0
	9022	G703
	9023	G79, G908, G80, X0, Y0
	9024	G408, P-.25
	9025	G403, P.5
	9026	G410, P-2.25
	9027	G703

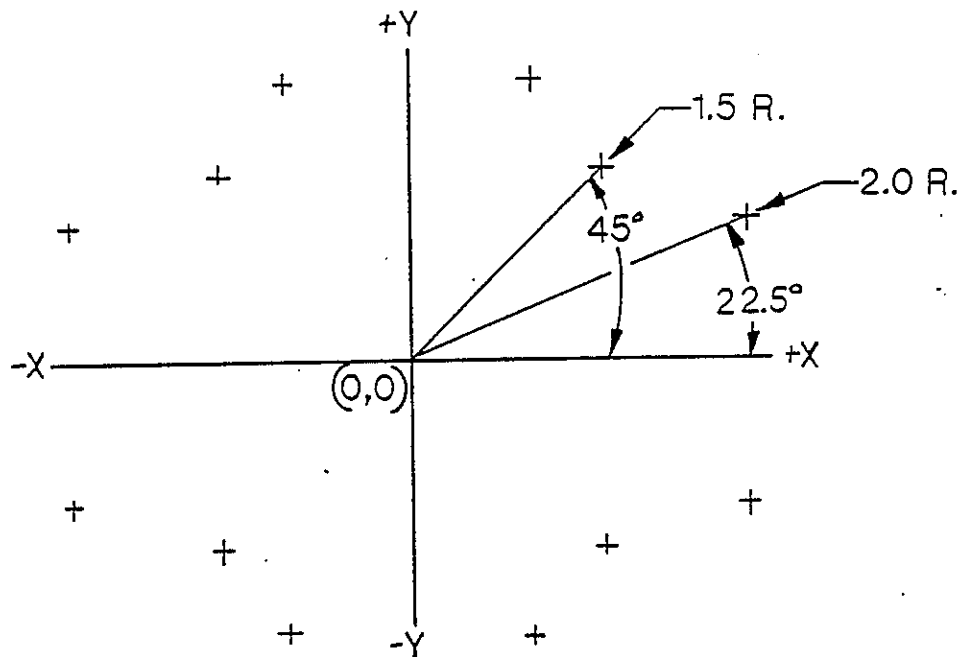


Figure 17.10 BOLT HOLE PATTERN

Since this example has more than one bolt hole pattern, the bolt hole routine is placed in subroutine #3.

```

2001 Activate Tool #10, Tool Change
2002 Selects a feedrate of 10 IPM; rapids tool to Floating Zero and
      turns ON rapid feed and coolant
2003 Sets R-Plane
2004 Set Z depth
2005 Set Dwell time in seconds
2006 Start Spindle
2007 Enable Drill
2008 Establishes the radius parameter for the bolt hole with G303
2009 Establishes the starting angle parameter (G310) for the bolt
      hole circle
2010 Selects the counterclockwise direction (G3) and calls up the
      subroutine twice (G820, L2). This block also shuts OFF the
      canned cycle, spindle, and coolant.
2011 Returns tool to Floating Zero

9022 Labels this as subroutine #3
9023 Turns the canned cycle on (G79) prior to running the bolt hole
      routine of 8 holes (G908) and then shuts the canned cycle off
      (G80) before returning to the center of the routine ready for
      the next loop.
9024 This block decreases the depth by .25 inch.
9025 This block changes the parameter (G403) by adding 1/2 inch
      to the radius
9026 Changes the starting angle (G401),(P-2.25), by subtracting
      22.5° from the original starting angle.

```

17.11 PARTIAL BOLT HOLE ROUTINE

BLOCK #	BLOCK ENTRY INFO-PROGRAM
3001	G111, M06
3002	G98
3003	G310, P7.5
3004	G300, P.1
3005	G306, P-.2
3006	G307, P.1
3007	G308, P-.75
3008	S500, M3, M8
3009	F10, G83, G0
3010	G804, L4, G80, M5, M9
3011	G100, G98, G303, P0
9028	G704
9029	G3, G79
9030	G303, P2
9031	G901, G80, X0, Y0
9032	G410, P2
9033	G704

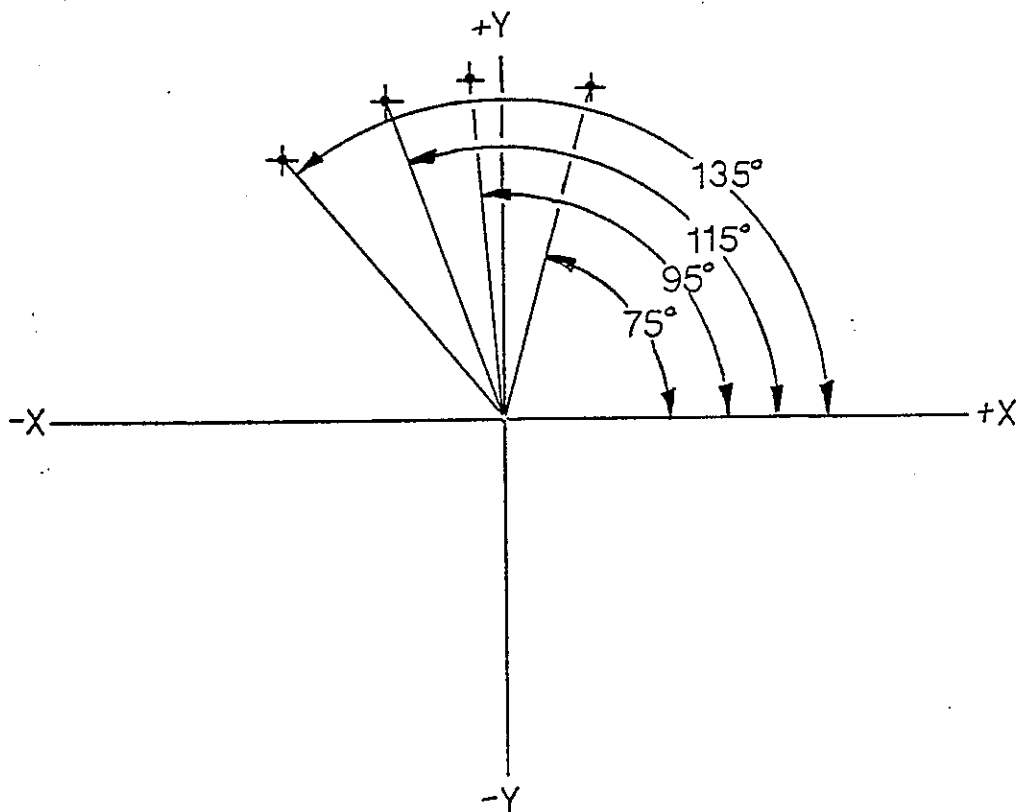


Figure 17.11 PARTIAL BOLT HOLE PATTERN

This example is very similar to example 17.10, only in this case just part of a bolt hole routine is needed. In this case, each hole is considered to be a bolt hole of one hole and the starting angle is simply shifted the required 20° each time the subroutine is repeated. This is accomplished by G410, P2 in Block 9031.

3001 Activates tool #11, Tool Change
 3002 The tool is brought to the center of the bolt hole routine
 3003 The starting angle is set at 75° (G310, P7.5)
 3004 Sets R-Plane
 3005 Sets Z 1st depth
 3006 Sets Z increment
 3007 Sets Z final depth
 3008 Starts Spindle
 3009 Enables peck drill cycle
 3010 Subroutine #4 is looped on 4 times (G804, L4)
 3011 Rapid to Floating Zero

17.12 FACEPLATE

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO-PROGRAM</u>
4001	G112, M06, G306, P-.25
4002	F10, G0
4003	G301, P2
4004	G302, P4
4005	G303, P1
4006	G300, P.1
4007	G308, P-.25
4008	G99, S1200, M3, M8, X2.25, Y3.625
4009	G86
4010	G2, G42, G26, G40, G80
4011	G0, X.5, Y4, G122 M06
4012	G301 P3.5
4013	G302, P6.25
4014	G305, P1.25
4015	G308, P-.15
4016	G0, X2.25, Y3.625
4017	S600, M3, M8, F5
4018	G85
4019	G27, G80, M5, M9
4020	G100, G98, G303, P0

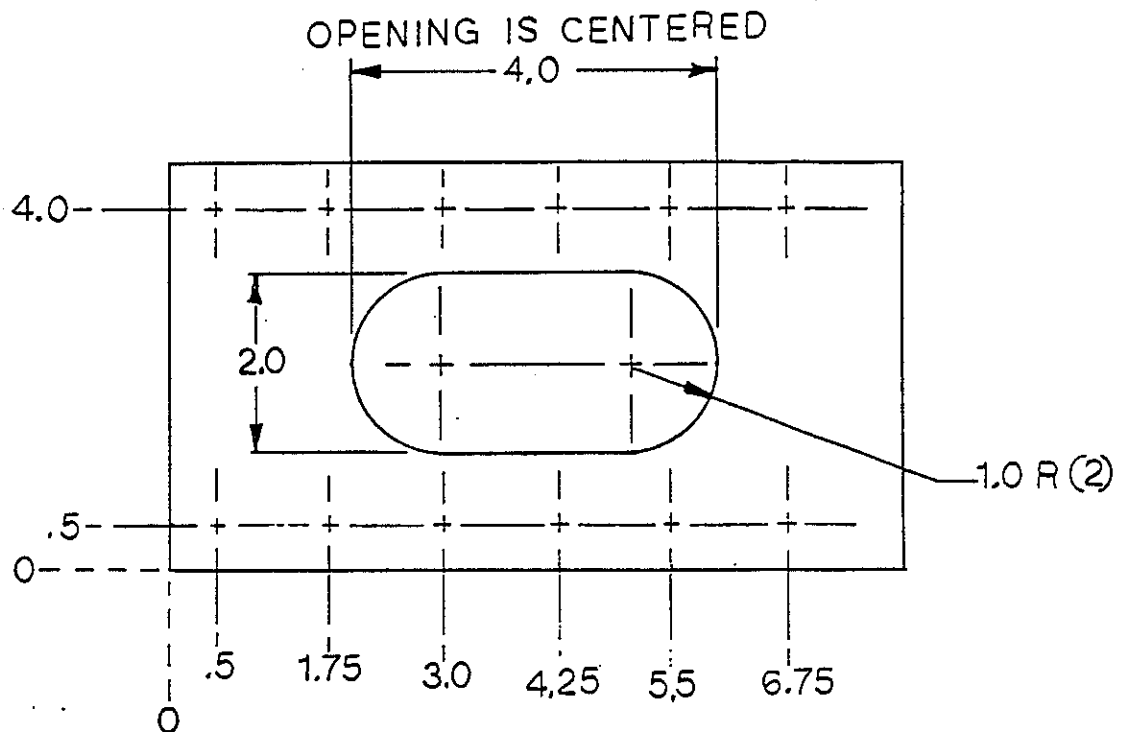


Figure 17.12

FACEPLATE

The faceplate program uses a combination of the symmetric hole drilling autoroutine (G27) and the finish rectangle autoroutine (G26). Note that both of these routines operate from the center of the workpiece, and that the dimensions entered into the parameters via G3XX code and are not dependent on the location of the Floating Zero.

```

4001 Activates Tool #12, Tool Change, and sets Z first depth
4002 A feedrate of 10 IPM is established
4003 X-pocket dimension
4004 Y-pocket dimension
4005 Radius of Corner
4006 Sets R-Plane
4007 Sets Z depth
4008 Positions the tool to the center of the workpiece and turn ON
    the spindle and coolant
4009 Enables Mill cycle
4010 Direction of cut and right cutter compensation is turned ON to
    give an inside cut and the G26 autoroutine is activated. Then
    cutter compensation is turned OFF. Z rapids to R-plane and
    Mill cycle is cancelled.
4011 Spindle and coolant are shut OFF and a program stop (M6) commanded
    to allow for a tool change
4012 Set X-dimension
4013 Set Y-dimension
4014 Set incremental distance between holes
4015 Set Z-depth
4016 Rapid to center of hole pattern
4017 Start spindle and coolant, set feed at .5 IPM
4018 Enable bore cycle
4019 Bore all holes, rapid Z to R-Plane and cancel Bore Cycle
    turn OFF spindle and coolant
4020 Rapid to Floating Zero
  
```

17.13 POCKET CLEARING ROUTINE FOR CIRCULAR POCKETS OF VARIED SIZES

BLOCK #	BLOCK ENTRY INFO-PROGRAM
5001	G114, M06
5002	G303, P.25
5003	G300, P.1
5004	G99, G3, F10, G0, X-1, Y.75
5005	G308, P-.15
5006	S2000, M3, M7
5007	G805, L4, G80, M5, M9, G100, G98, G303, P0
9034	G705
9035	G85, G306, P-.15
9036	G0, X1.75
9037	G86, G1, G307, P.1
9038	G305, P.1
9039	G41, G28, G40, G99, G408, P-.1
9040	G80, G403, P.125
9041	G705

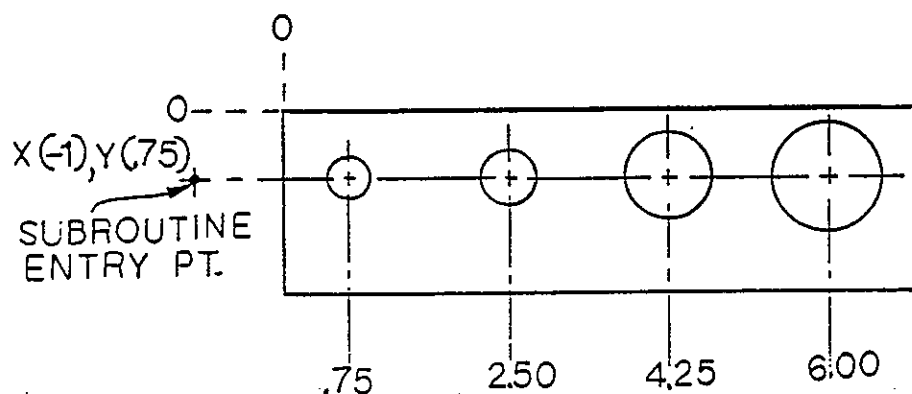


Figure 17.13

POCKET CLEARING ROUTINE-CIRCULAR POCKETS

This program uses a subroutine to do the circular pocket clearing (G28) as well as for changing the radius and depth of the pockets with the G4XX. Note that the cutter compensation is turned ON and OFF in the subroutine and therefore is OFF when going from the program to the subroutine and vice versa.

5001 Activates Tool #13, Tool Change
 5002 Radius parameter for the first pocket
 5003 Selects the feedrate, causes a rapid move to the required start position of the subroutine, and establishes direction of cut
 5004 Sets R-plane
 5005 Sets Z depth
 5006 Starts spindle and turns ON coolant
 5007 The subroutine is looped 4 times to accomplish all 4 pockets, the spindle and coolant are shut OFF, and the tool returned to the Floating Zero

17.14 THREE DIMENSIONAL CONTOURING

BLOCK #	BLOCK ENTRY INFO-PROGRAM
6001	G114, M06
6002	G300, P.3
6003	G1, X1.25, Y-1.1708
6004	G99, F1, S1500. M3, M7
6005	G1, Z0
6006	G19
6007	G03, R1.5, J-2, K-1.25, Y-2.8292, Z0
6008	X2.
6009	G02, R1.5, J-2, K-1.25, Y-1.1708, Z0
6010	X2.75
6011	G03, R1.5, J-2, K-1.25, Y-2.8292, Z0
6012	G99, G0. X5
6013	G124, M6, M3, M7
6014	G301, P2.
6015	G302, P1.6584
6016	G303, P0
6017	G308, P0
6019	G86
6020	G3, G42, G26, M5, M9, G40, G80, G99
6021	G100, G98

Tool #14 is a 1" ballnose
 Tool #24 is a 1" end mill

Figure 17.14

THREE DIMENSIONAL CONTOURING

```

6001 Positions the tool to the start of the cut as well as
    to establishing R-Plane, feedrate, spindle speed, starting
6005 spindle and turning ON coolant
6006
    to Accomplish the circular rise in the YZ-plane (G19)
6011
6012 Rapid Z to the R-Plane and position X for a tool change
6013 Tool Change
6014
    to Set pocket parameters and Z-depth
6017
6018 Position to center of pocket
6019 Enable milling cycle
6020 Finish the surface, stop spindle, turn OFF coolant, cancel
    Milling Cycle, Rapid Z to the R-Plane
6021 Rapid all axes to Floating Zero

```

It should be noted that this program will accomplish only a very rough cut of the part. Program 7 offers a method for accomplishing a finish cut on this part.

ALTERNATE PROGRAM

The tool for this program would be a ballnose once again, but would typically be a tool of half an inch or less. This program will make 100 passes at .02 inch increments across the part, but could be altered to do any number of passes by simply changing the X-axis step command in blocks 9044 and 9046, and the number of subroutine loops in block 7.

<u>BLOCK #</u>	<u>BLOCK ENTRY INFO-PROGRAM</u>
9042	G706
9043	G19
9044	G03, R1.5, J-.8292, K-1.25, Y-1.6584, Z0
9045	X.02
9046	G02, R1.5, J-.8292, K-1.25, Y0, Z0
9047	X.04
9048	G706
7001	G114, M06
7002	G300, P.3
7003	G0, X.8, Y-1.1708
7004	G99, F10, S1500, M3, M7
7005	G1, Z0
7006	G806, L50, G99, G17
7007	G0, X5
7008	G124, M6, M3, M7
7009	G301, P2.
7010	G302, P1.6584
7011	G303, P0
7012	G308, P0
7013	G0, X2, Y-2
7014	G86
7015	G3, G42, G26, M5, M9, G40, G80, G99
7016	G100, G98

17.15 STUDENT PROGRAM

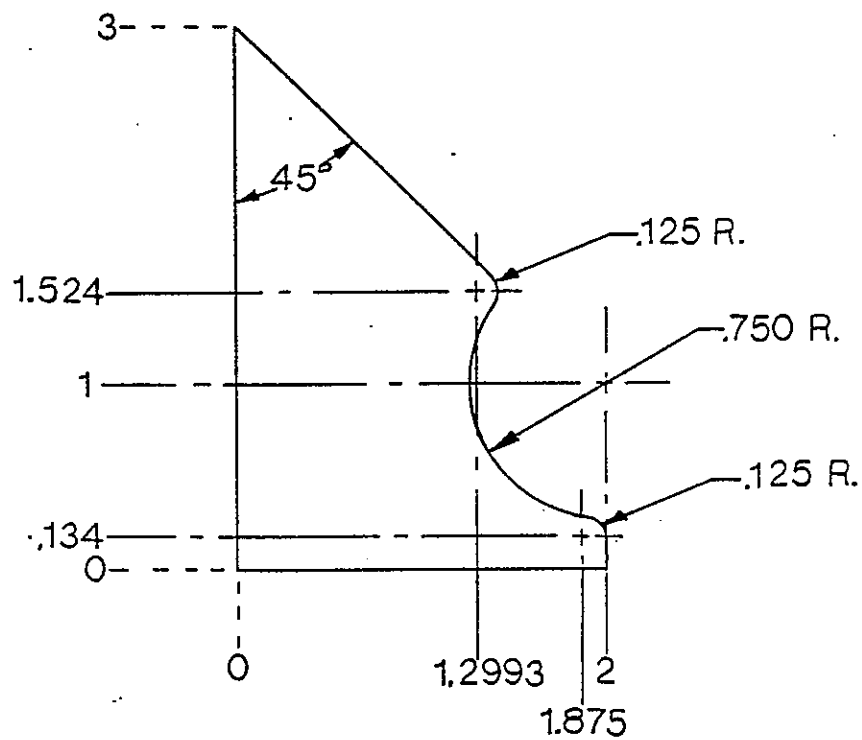


Figure 17.15

STUDENT PROGRAM

<u>BLOCK #</u>	<u>PROGRAM</u>
----------------	----------------

6001	
------	--

6002	
------	--

6003	
------	--

6004	
------	--

6005	
------	--

6006	
------	--

6007	
------	--

6008	
------	--

6009	
------	--

6010	
------	--

6011	
------	--

6012	
------	--

6013	
------	--

6014	
------	--

6015	
------	--

6016	
------	--

6017	
------	--

6018	
------	--

6019	
------	--

6020	
------	--

17.16 STUDENT PROGRAM FOR POCKET CLEARING

Write a program to clear a pocket with these dimensions

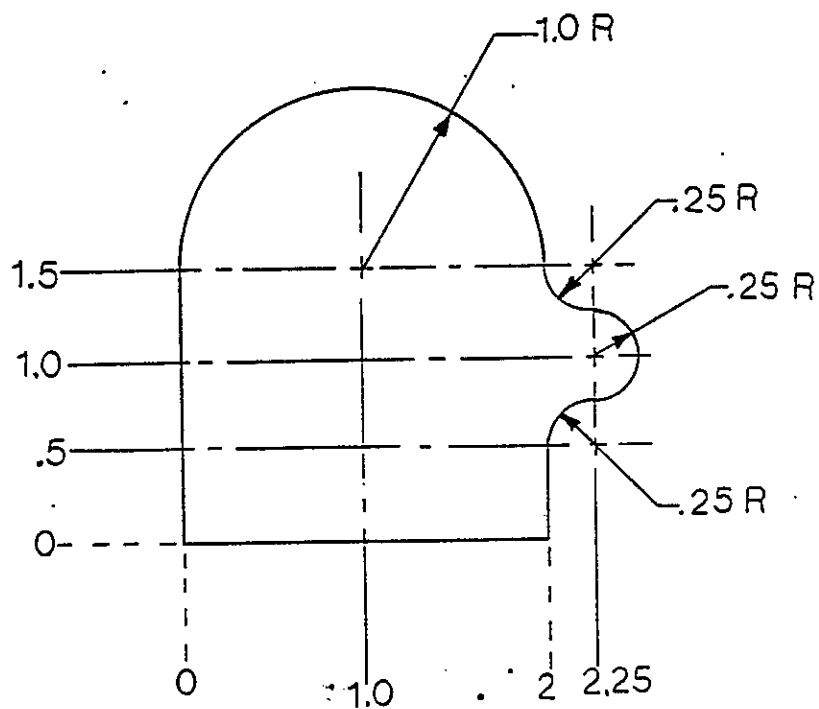


Figure 17.16

STUDENT PROGRAM FOR POCKET CLEARING

Write a program to clear a pocket of dimensions of Figure 17.16

<u>BLOCK #</u>	<u>PROGRAM</u>
----------------	----------------

7001	
------	--

7002	
------	--

7003	
------	--

7004	
------	--

7005	
------	--

7006	
------	--

7007	
------	--

7008	
------	--

7009	
------	--

7010	
------	--

7011	
------	--

7012	
------	--

7013	
------	--

7014	
------	--

7015	
------	--

7016	
------	--

7017	
------	--

7018	
------	--

7019	
------	--

7020	
------	--

SECTION XVIII

Prompt Operation

The help package is a system designed to allow the operator to write parts programs without a knowledge of EIA Codes (X, Y, Z, I, J, K, G, M, S, AND F). Instead of inputting these codes to generate a program the operator uses a menu or conversational screens to create the program. By selecting certain screens from the menus (MOVE, ARC, MIS., MILL, DRILL...ETC.), Standard EIA codes are generated for the operator. The generated EIA Parts program is also available for inspection using the display enter key.

The first step to familiarize yourself with the help package is to move through the screens using the flow chart (see drawing 1).

I. General Operations

A. Keys

1. Help Key

- a. Displays the main help screen
- b. Clears the event and operation counters (explained in Section II)
- c. It also sets the line and program counter. (except when coming from the tool data screen).
- d. All events & operations entered in this mode will be inserted at the beginning of the program.

2. Numeric Keys

The Numeric Keys "0 - 9", "+/-. " are used for entering programs and axis coordinate data. "0 - 9" are also used to select various modes and to move through the help menus.

3. Clear Entry Key

The Clear Entry Key is used to erase undesired information, if no information is at the curser position. The curser will move up one line.

4. Enter Key

- a. The Enter Key is used to accept entered information (axis coordinates, feedrates, etc.) or to move the curser down to the next line.
- b. The Enter Key is also used to return (RTN) to the previous screen if the curser is on the (RTN/Store) line.

5. Delete Key

The Delete Key is used to delete information at the current curser position.

B. Store-RTN line

If the operator wants to enter the information displayed on the screen into his parts program, the 0 key is pushed when the curser is at the store=0 position. If the operator does not want the displayed information stored in

his program, pushing the enter key will return to the previous screen and not store any data. (Note the help key may also be pushed to return to the main help screen and nothing will be stored).

III. Required Entries

1. Some lines require information to be entered in order to move the cursor to the next line (i.e. pushing Ent.) these lines will not allow the cursor to move to the next line until an entry is made.

Required entries are as follows:

- a. Prog.#, ABS-INC, INCH-METRIC (Start program screen)
 - b. Z FEED-RATE, 1st Z-DEPTH, Z-INCREMENT (MILL SCREEN)
 - c. CLEARANCE, Z-DEPTH, (Can-Cycles)
 - d. CUT WIDTH, FINISH STOCK, POCKET RADIUS, X-Y POCKET DIMENSIONS, CW-CCW, CUTTER COMP. OFF-ON (AUTOROUTINES)
 - e. PROG.-SUBPROG. #'S, EVENT-OP.#'s (DISPLAY-EDIT SCREENS)
 - f. PROG.#-EVENT# (RUN PROGRAM SCREEN)
2. The screen will not be stored until sufficient data has been entered. If this case occurs the cursor will back up to the requested entry, when the store function is selected.

These screens are as follows:

- a. Radius, center, and end point or angle on the arc screen.
(Note: if both polar and rect. coordinates are entered the rect. coordinates have priority and they will be stored. A combination of polar and rect. coordinates is also allowed.
- b. At least one axis on a line, move, or location screen.
- c. Polar radius without a start angle or end angle on the line, move, location screens.

IV. Illegal Instructions

1. Some entries are not allowed and the cursor will move to the line that is illegal when a store is attempted.

These lines are as follows:

- a. Z inc. = 0, Clearance less than Z-depth (CAN-CYCLES)
- b. CUT WIDTH=0, X-Y POCKET DIMENSIONS = 0 (AUTOROUTINES)

V. Display and editing programs or subroutines

- A. Once a program has been developed (or partially developed) using the help system, the operator can display or edit the program using the display/edit screen. The simplest form would be used to display the entire program or subroutine. Enter Prog.# and Event #1 then push 5-(Display/Edit program or Subroutine). The first event in the program will appear on the screen and

the curser will stop at the store=0, RTN=ENT line, to display the next event (or operation) press "0". Continue pressing 0 until the end of the program.

1. If you want to modify or delete any of the displayed information on the screen, use the CLR ENT KEY to back up to the desired line, make the change or deletion and use the Ent. Key to move back to the next operation or event.

1. If Ent. is pushed at the store=0, RTN=ENT line the same screen will reappear. (This can be used if during editing the operator forgets what the original event or operation was).

B. To display a specific part of a program the operator can select which Event # he would like to display or edit.

1. Displaying a tool # or event # that does not exist, will set the line counter to the end of the program and display the end of program screen.

2. Displaying or editing on the start program screen will not clear the program.

VI. Entering and Running a Simple Help Program. (Drawings 2 & 3)

To program in the Help Mode start by depressing the Help key. The first menu will appear on the screen. There are seven possible selections to choose from. Choose desired function and push its corresponding key. If we wish to create a new program the number 1 key is our selection. After pushing the 1 key a new menu will appear, labeled initialize program. Answer the questions on this screen and push the 0 key to store your entries. After the initialize program screen the control will automatically advance to the event screen. The event screen has the following 9 choices.

1. Move - The move screen is selected whenever you wish to position or move the machine without doing a cutting operation.
2. Cut - The cut screen is selected when a cutting operation is required. (ie: Milling, Drilling, Tapping, Boring)
3. Tool Change - This screen is selected whenever a tool change is desired.
4. Tool Call - This screen is used to tell the control to use a different tool in the offset table but do not stop and require a new tool to be put into the spindle.
5. Misc. - This screen is used to activate the various functions listed on the screen.
6. Modify Parameters - This screen is used when the programmer wishes to modify one the 100 programming parameters.
7. Modify Tool - This screen is used when you wish to modify one of the 24 tool offsets, in the tool offset table.
8. Scale, Rotate, Mirror Image - This screen is selected when a program is to be scaled or mirrored.
9. Comp. Subroutines - Used to selectively cutter comp. individual subroutines. Must be used if scale factor, rotation or tool radius is changed in a program using milling subroutines.

The event selection one would normally make at this point in the program is a tool change. Push the 3 key and answer the questions on the screen, after storing the entries, the control will exit back to the event screen. and wait for the next selection. The next step in a normal program sequence would be a cutting event. The 2 key will select a cutting event and the cut screen will appear, it has the following nine cut types.

1. Mill - The mill cut type is selected anytime line, arc, autoroutine, corner rounding, circle generate functions are desired.
2. Drill - When drilling is desired this selection should be made. Bolt hole patterns are also programmed off this screen.
3. Drill Dwell - Same as drill but has a programmed dwell time at the bottom of each hole.
4. Drill Peck - Same as drill but Z uses a peck cycle.
5. Woodpecker Drill - Same as drill but Z uses woodpecker cycle.
6. Bore - Same as drill but Z feeds into and out of the part.
7. Bore Dwell - Same as bore but Z dwells at the bottom of the bore.
8. Tap - Same as drill but Z feeds in and out of the hole and the spindle automatically changes direction at the bottom of the part.
9. Subroutine - This selection is made when the cutting pattern has been programmed in a subroutine. The cut pattern can be either milling or drilling. The selection is made at the point in your program you wish the pattern to be executed at.

Select the cut type you desire and push the corresponding key. If our cut type was milling we would push the 1 key. After pushing the 1 key the mill screen will appear. The first part of the mill screen asks questions pertaining to the Z-axis. The bottom half of the screen has choices which will be used to describe the X-Y contour or shape of the part. The following selections are possible from the mill screen.

1. Misc. - Used to change feedrate, turn coolant on and off, turn spindle on and off, etc.
2. Line - This screen is selected whenever a line is to be programmed as part of the contour.
3. Arc - If a circle or arc is to be programmed this selection would be made.
4. Corner Functions - This screen is selected anytime you want to round a corner, generate a circle from 3 points, turn cutter comp. on or turn cutter comp. off.
5. Autoroutine - This selection would be made if you wanted to do one of the preformatted autoroutines. These autoroutines are the following:
 - A. Cir. Pocket Clear - Spiral pocket clear out to a specified radius. (9-11)
 - B. Cir. Finish Inside - Makes a circular finish pass. (9-3)
 - C. Cir. Finish Outside - Makes a circular finish pass. (9-3)
 - D. Rectangular Pocket Clear - Clears a rectangular pocket to a specified depth. (9-8)
 - E. Rectangular Finish Inside - Takes a finish pass around a rectangular pocket. (9-5)
 - F. Rectangular Finish Outside - Takes a finish pass around a rectangular pocket. (9-6, 9-7)
6. Subroutine - This selection would be made if your part was programmed in a subroutine and you wished to run the subroutine at this point in your program.

Using the above choices, any part contour can be programmed. The mill cycle will continue to repeat this contour until the Z parameters programmed on the top half of the screen are satisfied. Note how the square was entered in drawing 2.

If the desired cut type would have been one of the tapping, drilling or boring routines, we would have made selections 2 thru 8 instead of 1. The top half of the tapping, drilling and boring screens describes the hole depths. The bottom half of the screen is used to program the hole locations. The bottom half of the screen has the following choices:

1. Misc - Used to change feedrates, turn coolant on and off, turn spindle on and off, etc.
2. Location - X-Y position of holes to be tapped, drilled or bored.
3. Bolt Hole - If a bolt circle is to be tapped, drilled or bored at this location in the part program make this selection.
4. Subroutine - This selection would be made if your hole pattern was programmed in a subroutine and you wished to run the subroutine at this point in your program. (Drawing 3)

Following the flow chart of Drawing 2 note how the cutter comp. was turned on and off. The part being programmed in Drawing 2 is shown in Drawing 6. Drawings 5 and 6 give detailed explanations on how to turn cutter comp. on and off relative to any part shape. Page 18-7 gives another example of how to turn cutter comp. on and off.

The flow charts of Drawings 3 and 4 show how to program a Bolt Hole pattern and a random hole pattern using subroutines. The things to note on Drawing 3 are the rapid non-cutting moves from the tool change position to the center of the bolt hole. If these moves were not done the bolt circle would be drilled about the tool change position. Another point to notice is the use of the ENT key to back up to the previous screen if your next choice is not displayed. Drawing 4 shows how to program a series of holes in a subroutine. The hole pattern is nothing more than a series of XY moves in either rectangular or polar coordinates. The key points to remember when programming a subroutine is to start and end it with the same subroutine number. The floating zero for the subroutine is automatically set at the position of the machine when the subroutine is called.

Drawing 7 shows flow charts of the 3 things that must be done before a program can be run. They are Gridding the machine, setting the part floating zero, and setting the tool offsets for the part. After the 3 steps have been done the part program can be run. (Drawing 8) To run a program push the Help key, select number 6, Run Program, enter the program number to run, skip the tool number by pushing the ENT key and type 1 for the Event number. After verifying the data is correct push cycle start and the program will start to run. The above sequence will start the program at the beginning. If the tool number is entered the program will start running at the Event number of the specified tool. If you wish to start a program at the tool change you would enter the desired Tool number and a 0 in the Event number.

VII. Special Features

A. Corner Rounding

1. Round last corner.

Example #1 line to line corner

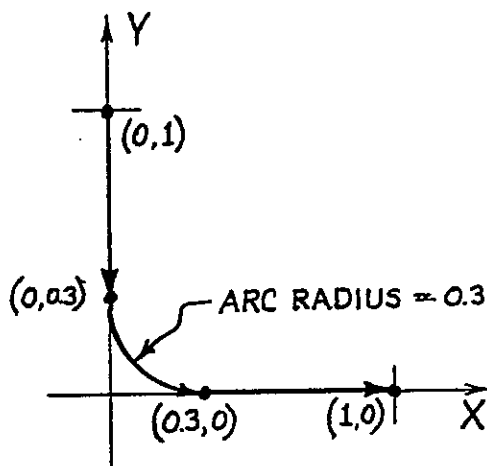
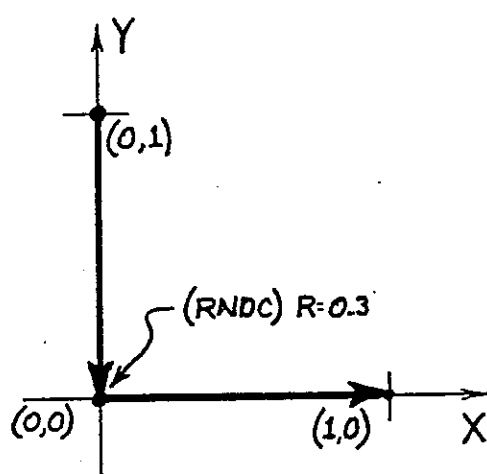
Move X00.0000 Y01.0000

Move X00.0000 Y00.0000

Round last Corner R00.3000

Move X01.0000 Y00.0000

An arc will be inserted in the corner with radius=0.3.



Example #2 arc to arc corner

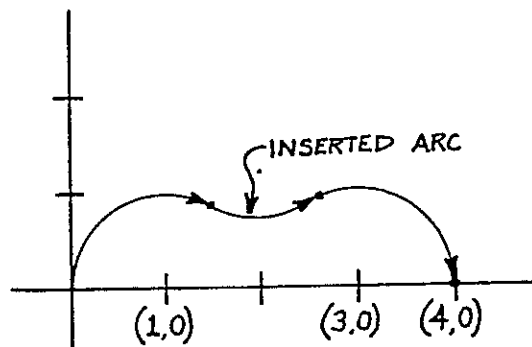
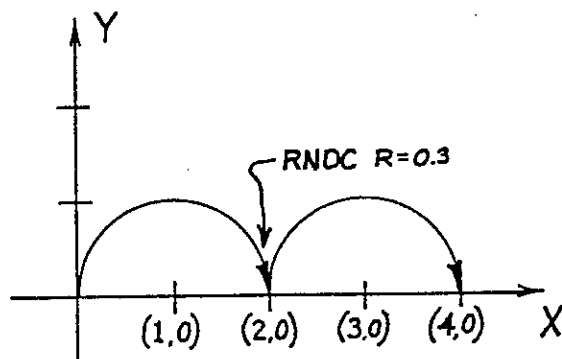
Move X00.0000 Y00.0000

ARC. CW, Radius R=1, CENTER=X01.0000 END PT.=X02.0000
Y00.0000 Y00.0000

Round last Corner R00.3000

ARC. CW, Radius R=1, CENTER=X03.0000 END PT.=X04.0000
Y00.0000 Y00.0000

An arc of .3 radius will be inserted tangent to the two circles.



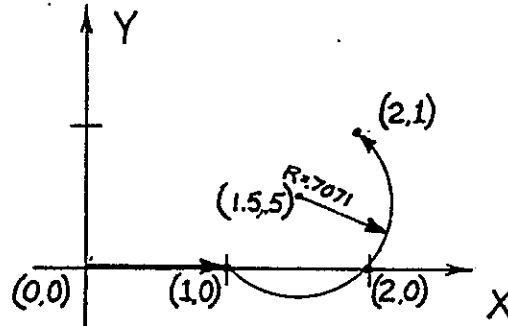
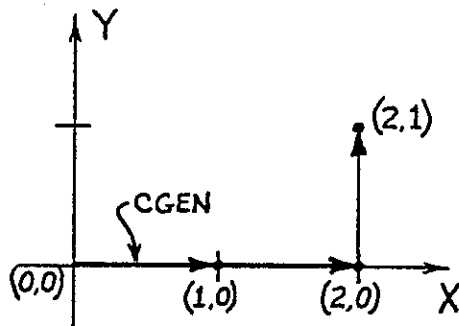
B. Round last corner can also be used for line to circle and circle to line corners in the same manner as line to line and circle to circle.

C. Generate circle from next 3 points

This is used to generate an Arc when the radius and/or center is not known.

The 3 points given after the corner generate command (CGEN) are used to define this arc. The first point is the starting pt. of the arc the third point is the end pt. of the arc.

Example #3 Generate Arc from 3 pts.



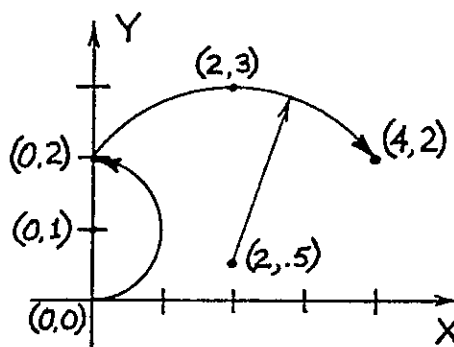
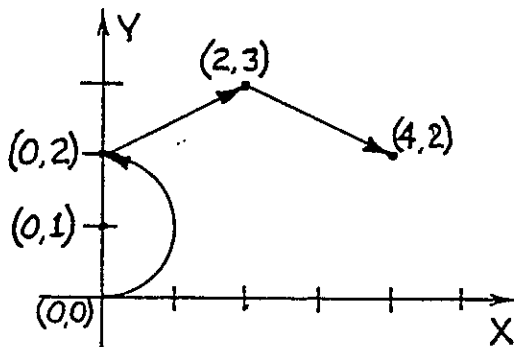
Entered Program

```
X0      Y0
(CGEN)
X1      Y0
X2      Y0
X2      Y1
```

Final Program

```
X0      Y0
X1      Y0
G003 G017 R.7071 I 1.5 J.5 X2 Y1
```

Example #4 Generate Arc from 3 pts.



Entered Program

```
G003 G017 R1 I0 J1 X0 Y2
(CGEN)
*X0      Y2
X2      Y3
X4      Y2
```

Final Program

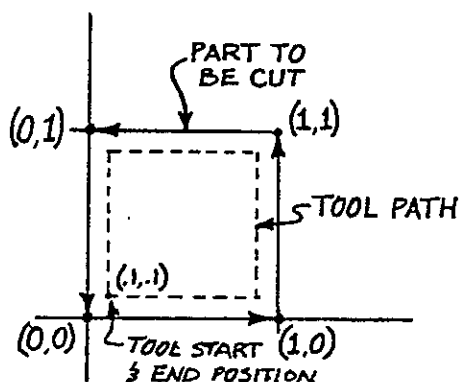
```
G003 G017 R1 I0 J1 X0 Y2
*X0      Y2
G002 G17 R2.5 I2
J.5 X4 Y2
```

(Note *Start pt. and second arc must be entered after (CGEN) for correct arc generations. (CGEN) cannot be used with arc's as pts. to generate an arc. If the pts. generate a radius greater than 99.999 the (CGEN) is ignored.)

D. Enter-Exit Cutter Compensation

This is used to position the tool such that the part being made is not cut into at its starting and ending points when cutter compensation is being used. An example is the best way of showing how this feature works.

Example #5 Enter-Exit Cutter Compensation.



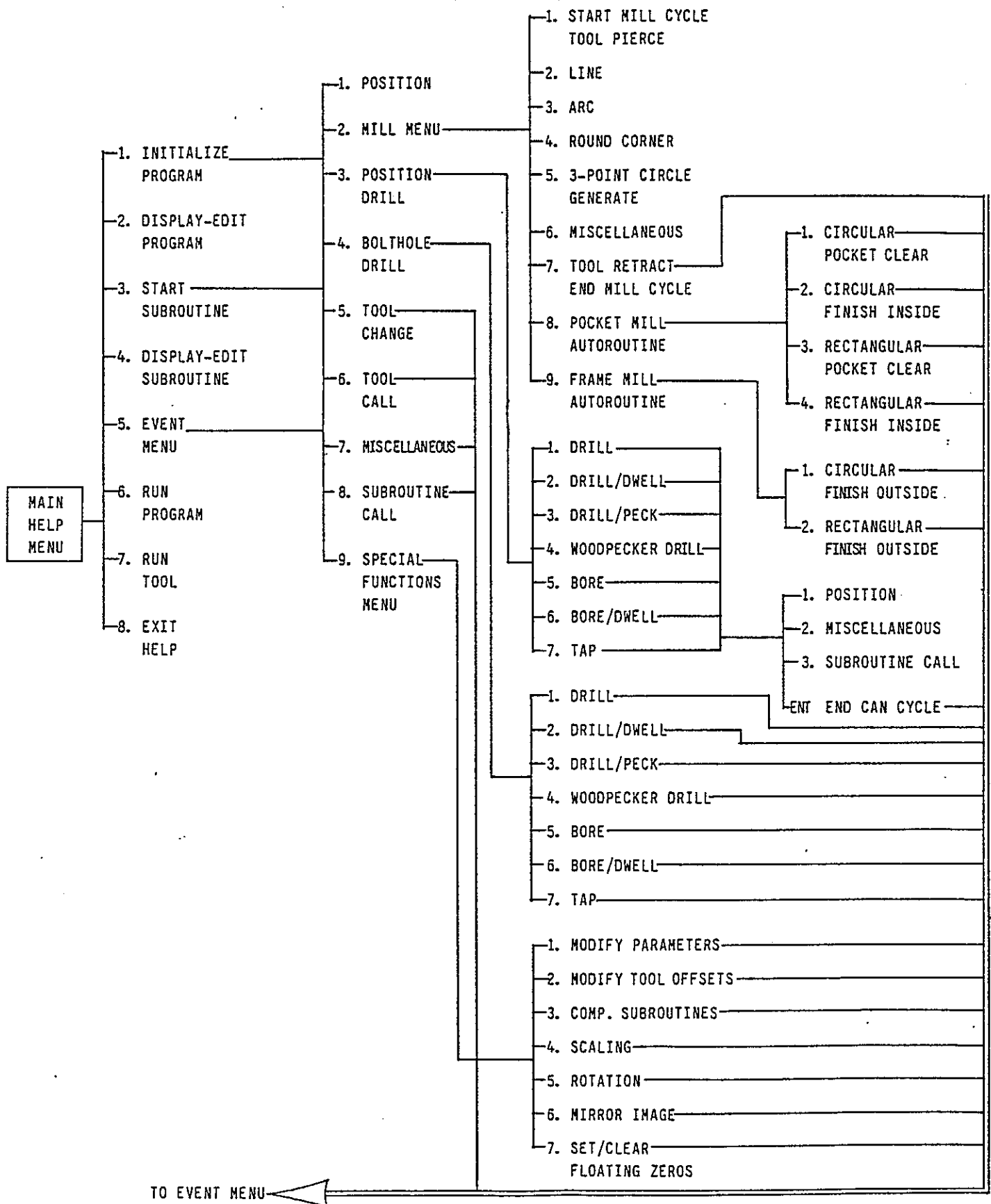
```

Tool #2 X5 Y4 Tool Change Position
Enter C.C. (LEFT CUT) X0, Y1
Move X0 Y0 1st. pt.
Move X1 Y0 2nd pt.
Move X1 Y1 3rd pt.
Move X0 Y1 4th pt.
Move X0 Y0 5th pt.
Exit C.C. (C.C. OFF) X1, Y0
    
```

When entering cutter comp. with a tool radius=.1 and giving pt. X0, Y1 the tool will be positioned at X.1, Y.1 when Z feeds down. This prevents the line (X0, Y1) to (X0, Y0) from being cut into. The point X0, Y1 is used only to calculate the pt. which the tool will feed down at and X and Y axis will not move to this pt. Any pt. on the line (X0, Y1) to (X0, Y0) can be given. When exiting cutter comp. with a tool radius=.1 and giving pt. X1, Y0 will cause the tool to stop at (X.1, Y.1) preventing the line (X0, Y0) to (X1, Y0) from being cut into. Any pt. on this line can also be given.

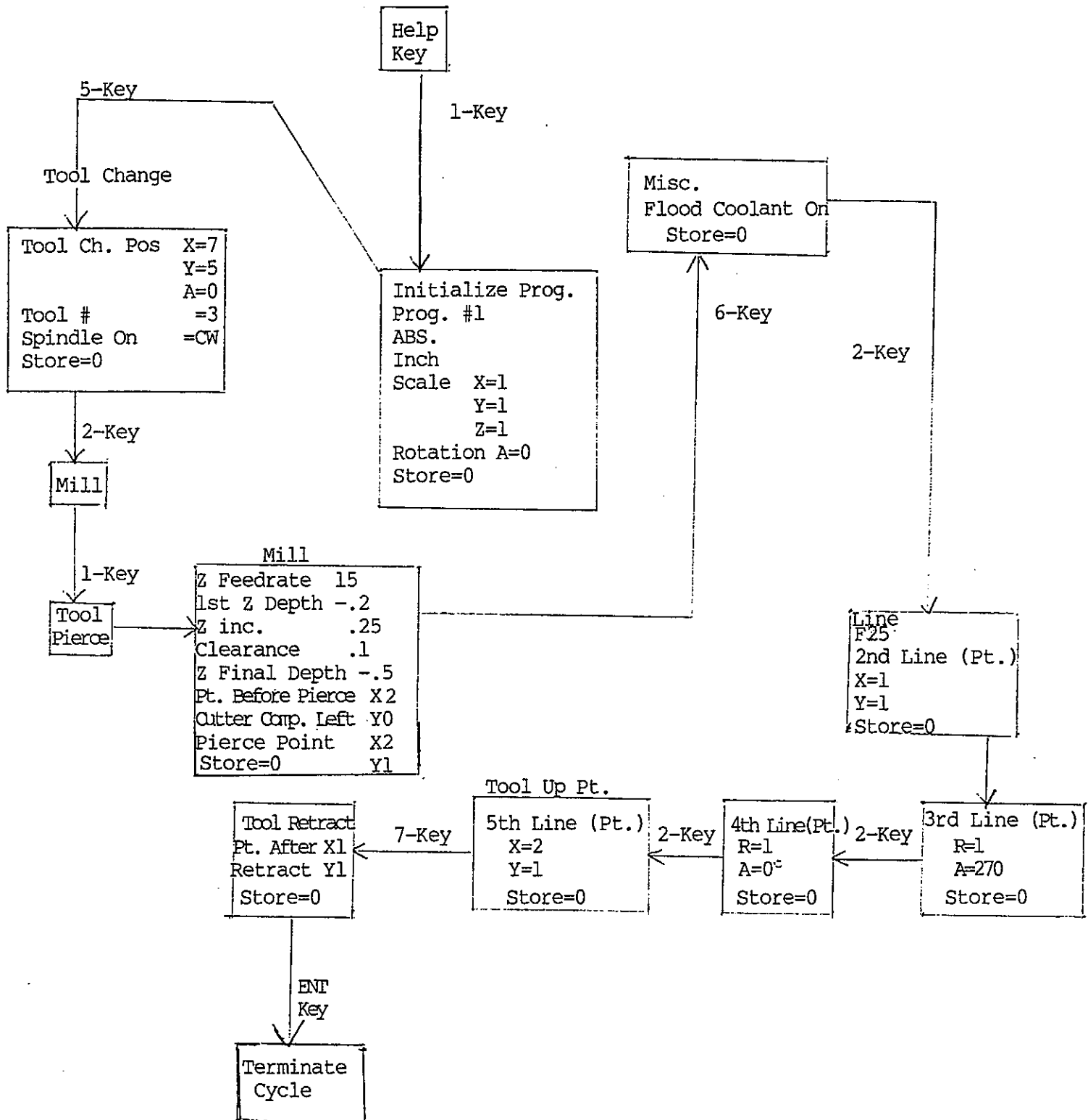
PROGRAMS NOT DEVELOPED ON THE HELP SYSTEM CANNOT BE DISPLAYED ON THE HELP SYSTEM.

HELP FLOW CHART



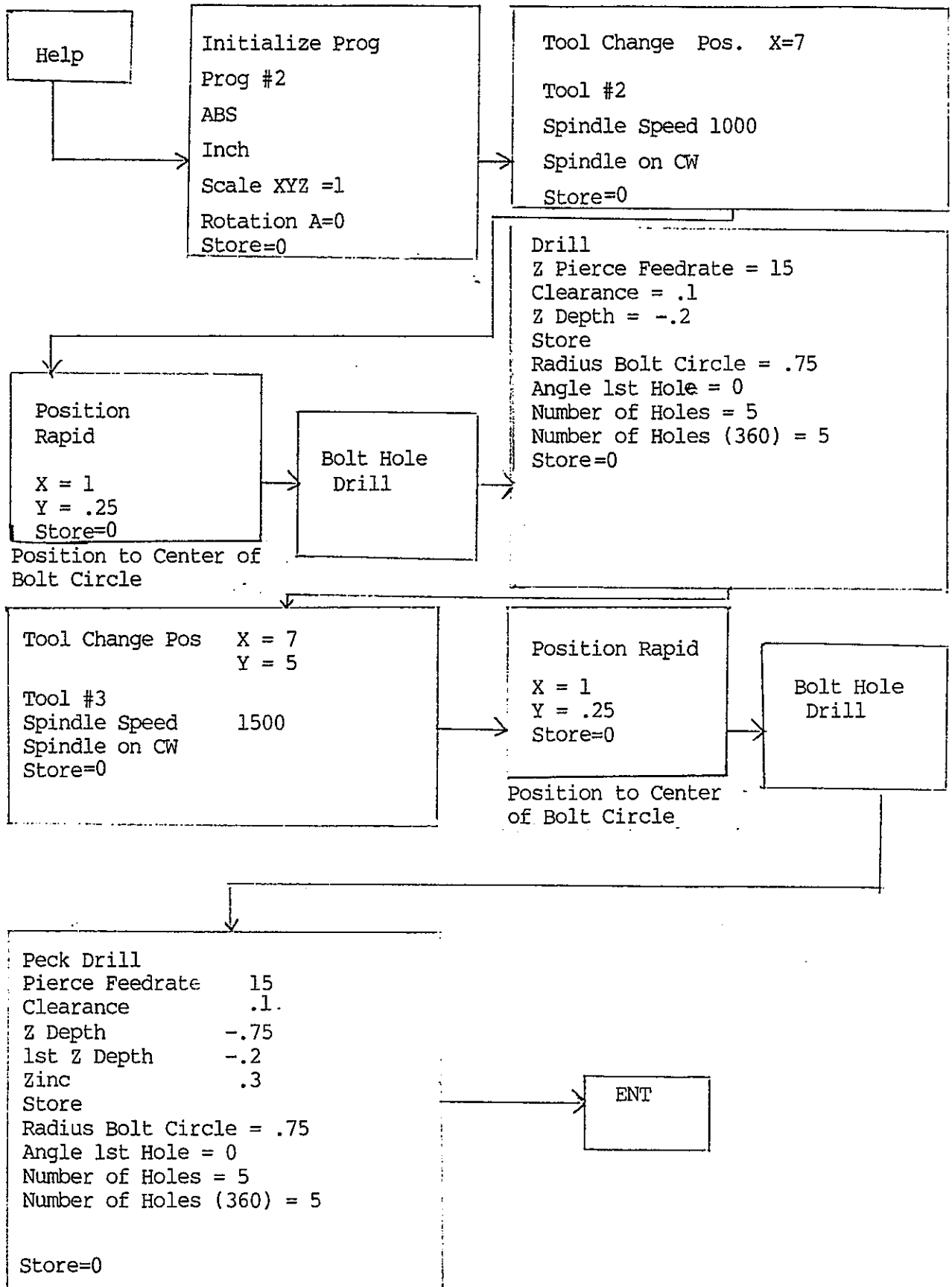
Flow Chart To Enter A Simple Program
(Mill a 1 inch square, .5 inch deep, with cc. and .25 tool)

Drawing 6

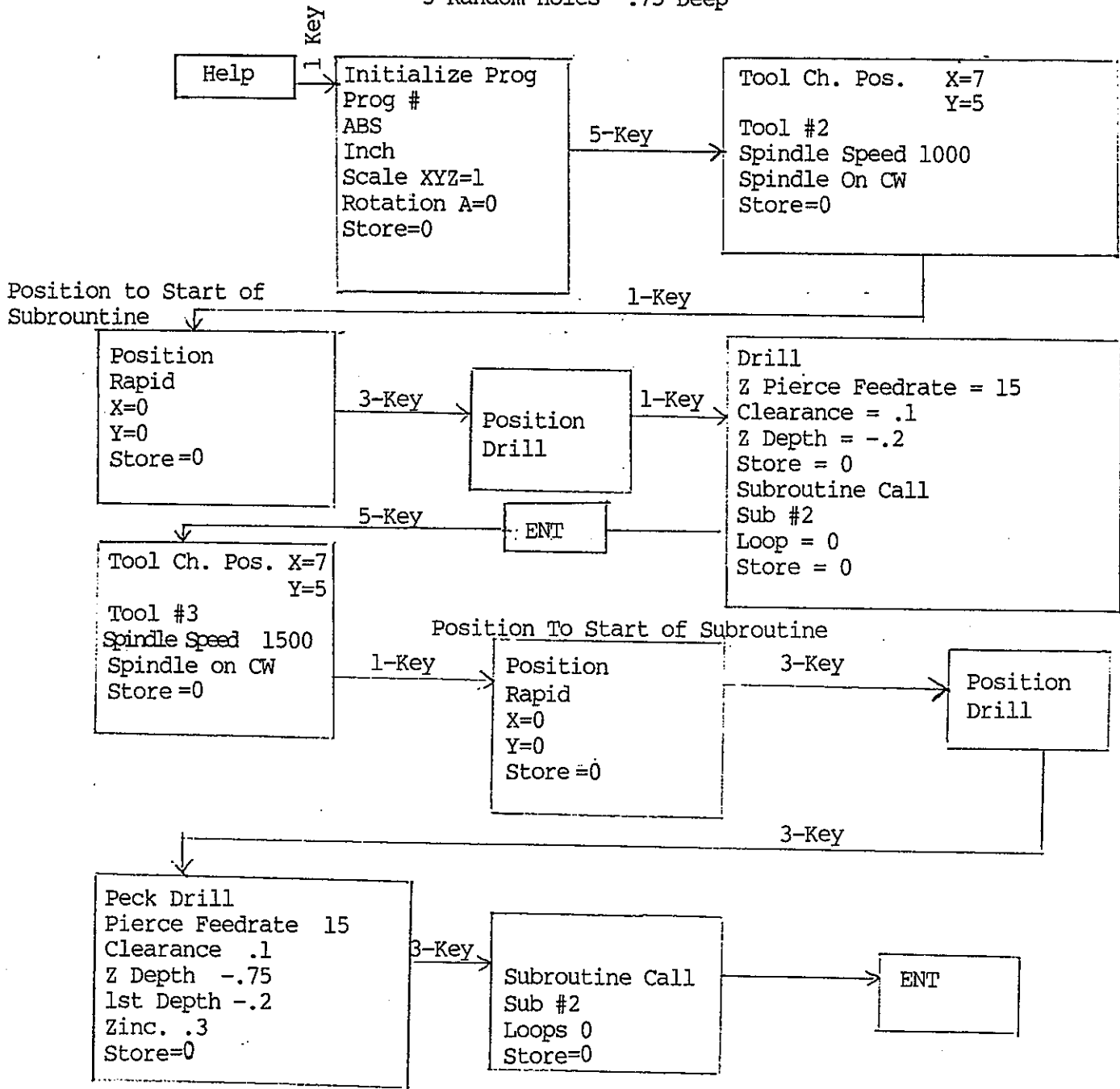


Drawing 2

Flow Chart To
Spot Drill and Peck Drill
5-Hole Bolt Circle

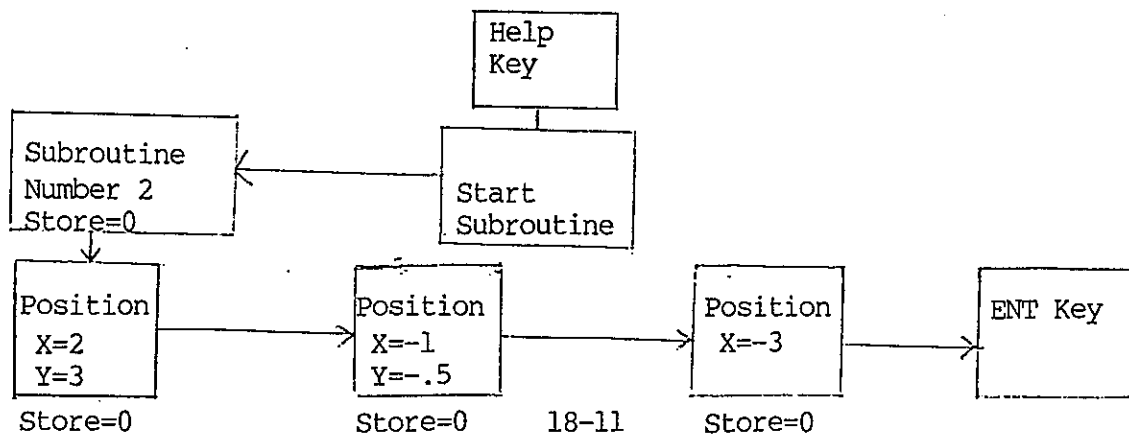


Flow Chart To
Spot Drill and Peck Drill
3-Random Holes .75 Deep

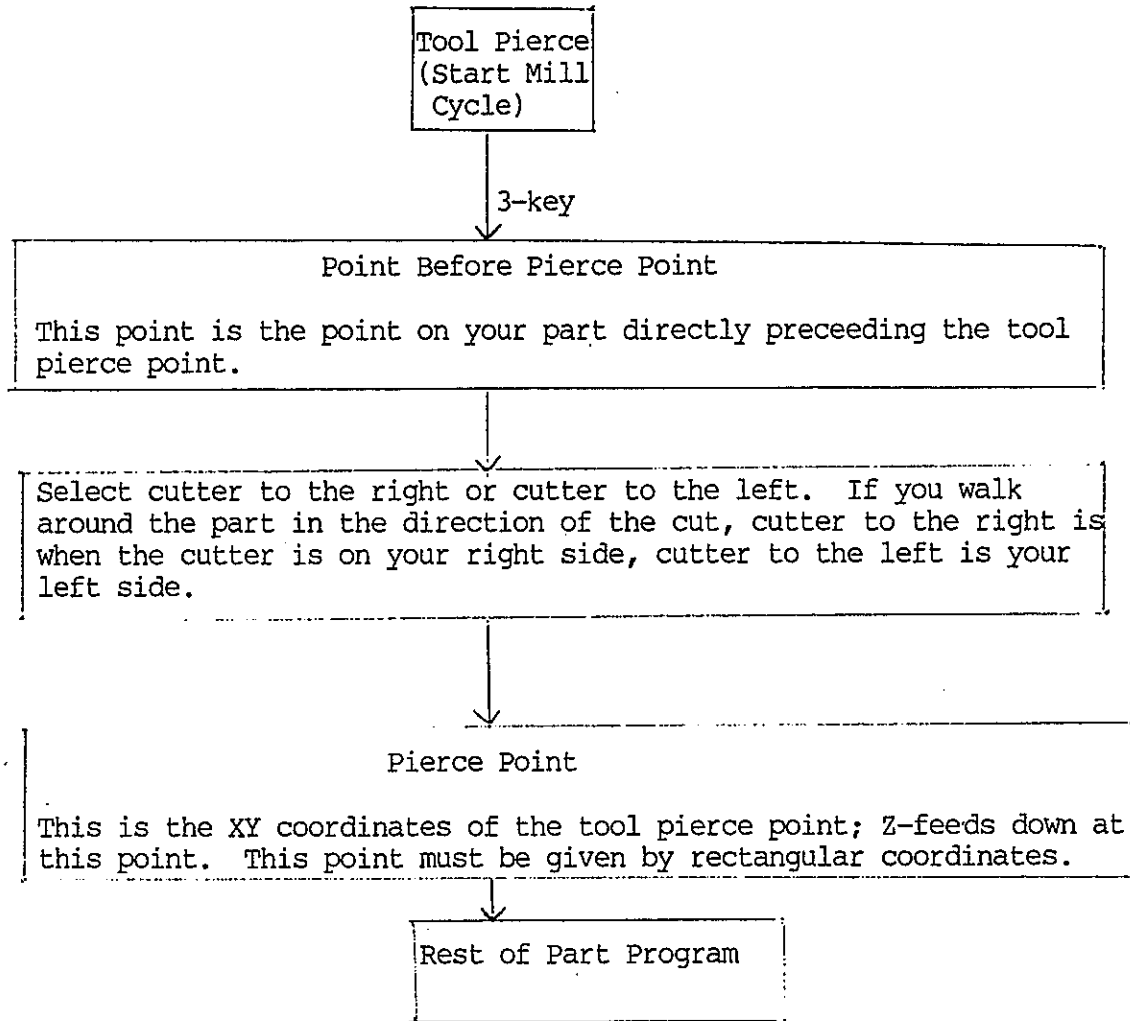


Enter 3-Hole Subroutine
for Drilling

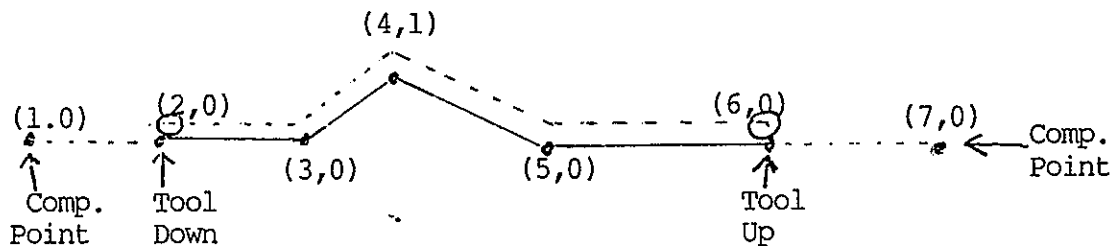
(All dimensions are relative to start of subroutine)



Enter Cutter Comp.



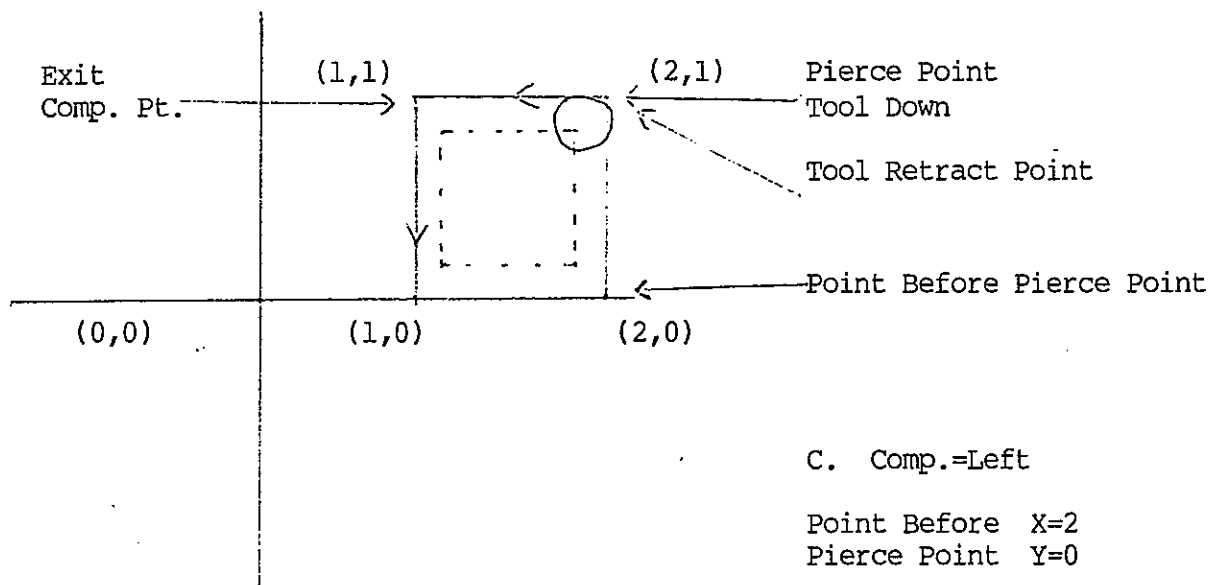
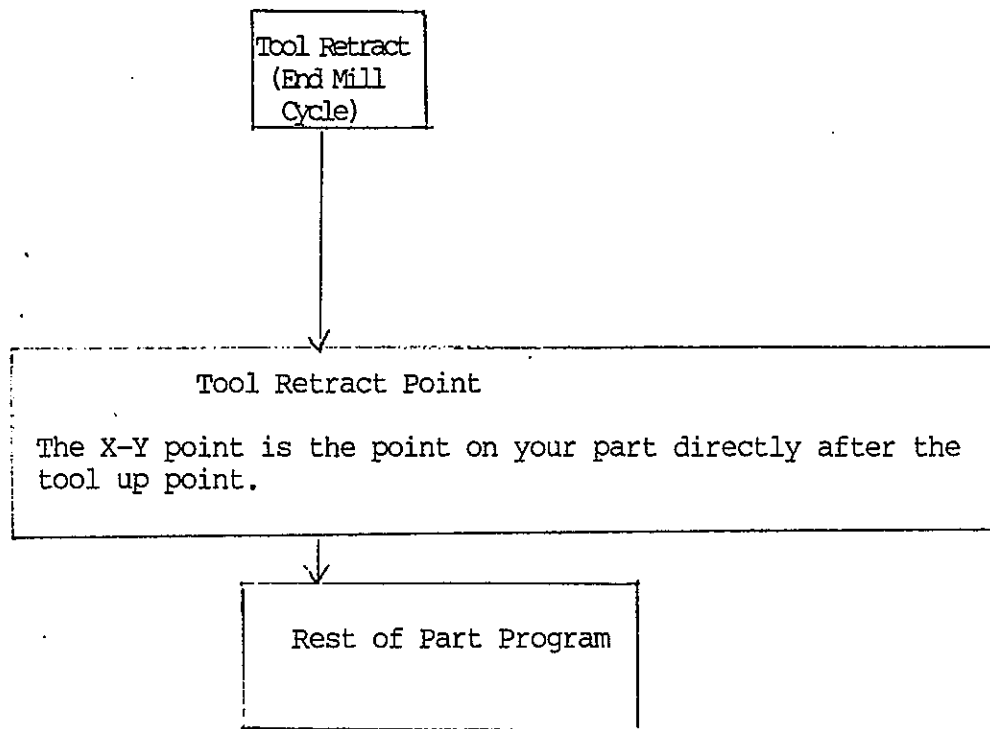
PART WITH OPEN CONTOUR



On the part with an open contour, if the comp. pts. lie on the same line as the first and last points of the part, the tool center will be over the start and end points of the part, when the tool pierces and retracts.

Drawing 5

Exit Cutter Comp.



C. Comp.=Left

Point Before X=2
Pierce Point Y=0

Pierce Point X=2
Y=1

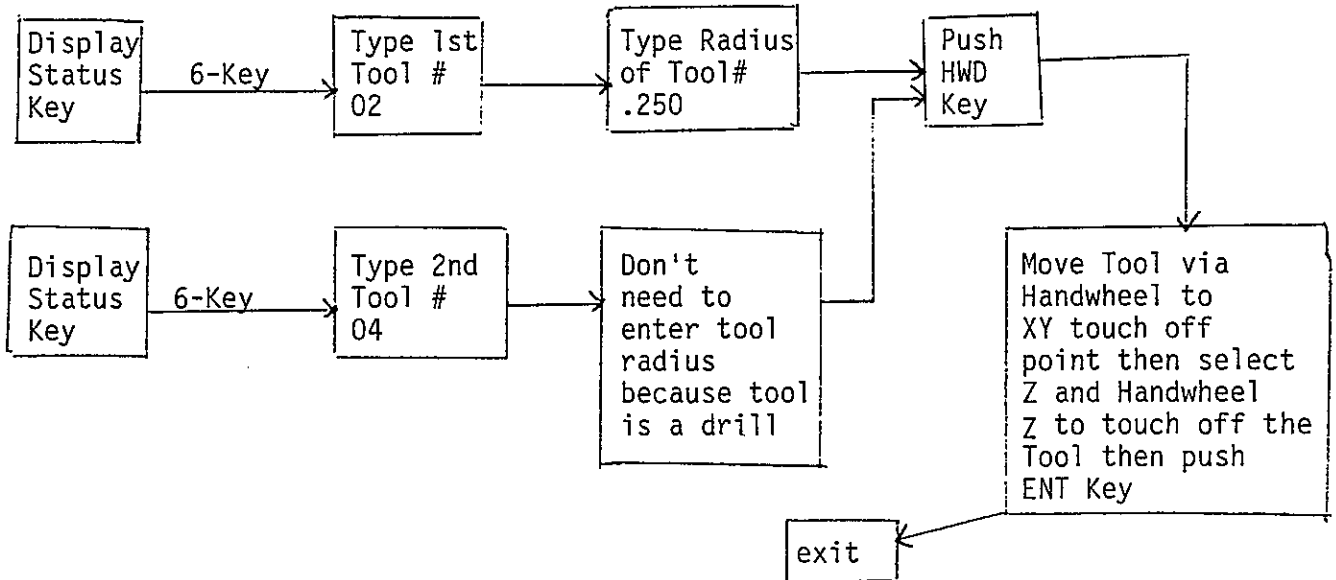
Program
Tool Retract (End Mill)
Point After X=1
Retract Point Y=1

Drawing 6

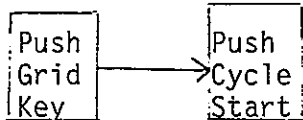
Things to do before a Program is Run

Set Tool Offsets for Tools used in the Program

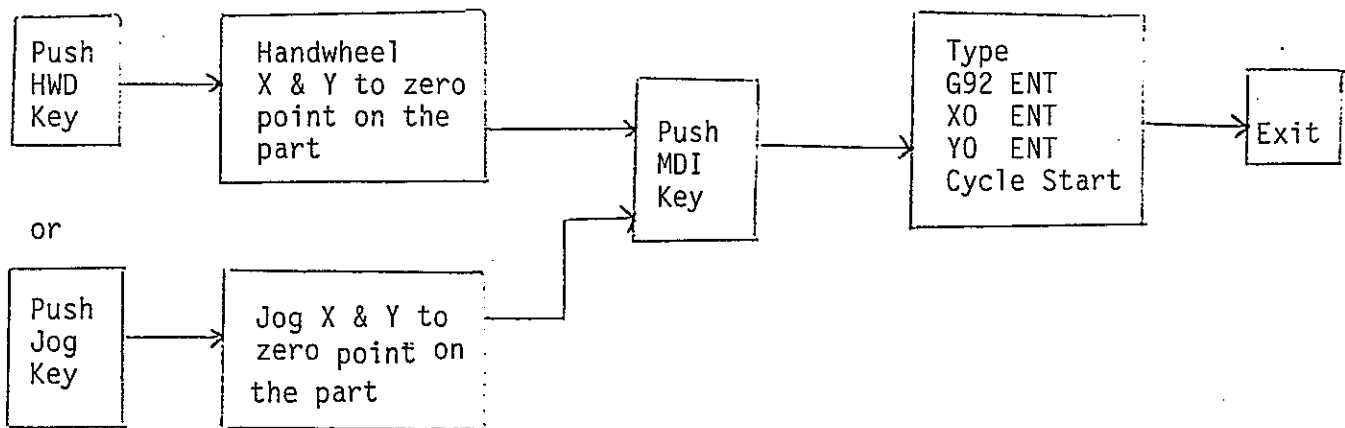
Example: Tool#02 is .500" end mill
Tool#04 is .125" drill



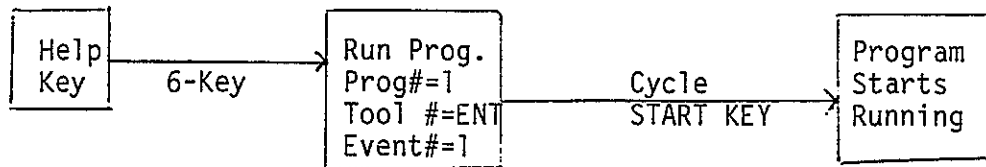
Grid Machine



Set Floating Zero to Part Program Zero Point

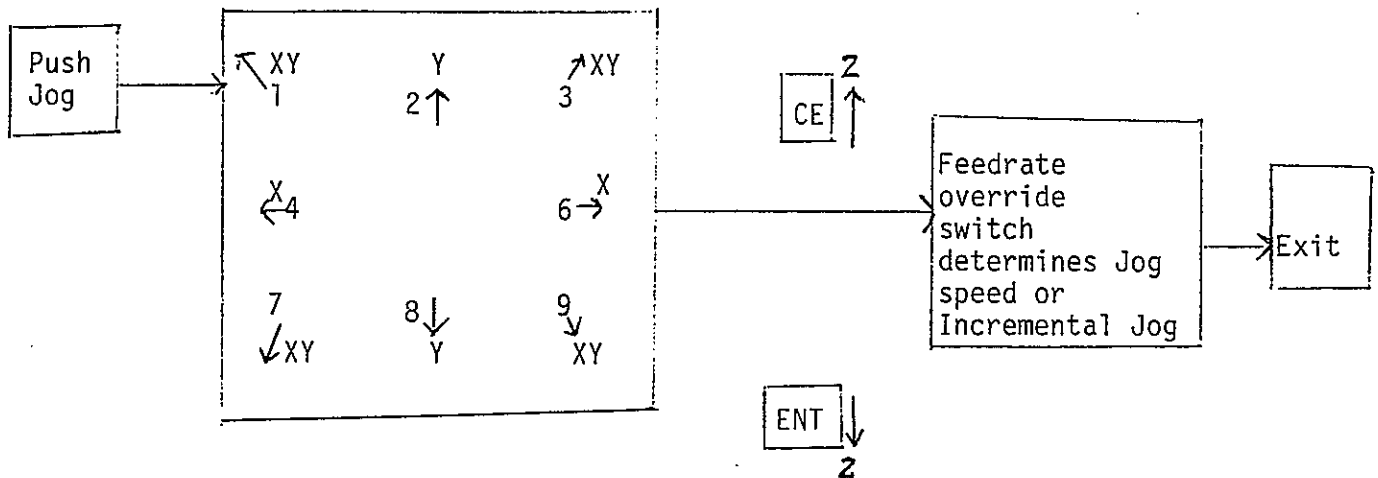


Run Program

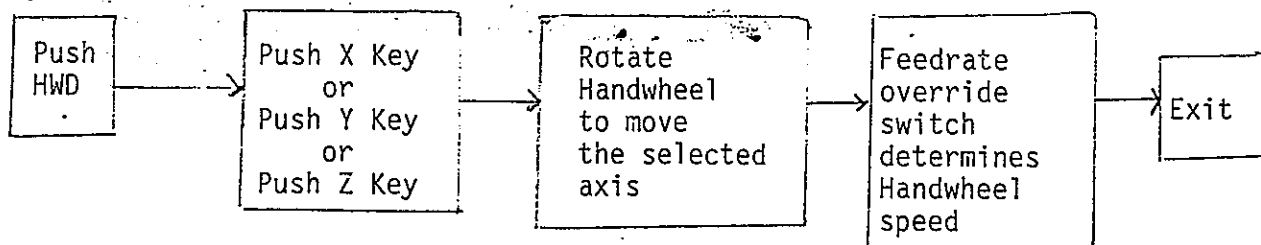


NOTE: If tool # is filled in, the Event # should be set to zero. The program will then start running at the specified tool number.

Jog Machine



Handwheel Machine



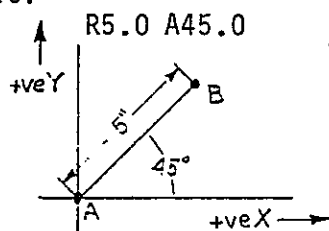
SECTION XIX

Polar Co-ordinate Programming.

In addition to the rectangular (or cartesian) co-ordinate programming modes, the CNC allows Polar (or angular) programming.

Linear Case: The user specifies an incremental radial distance from the start point and an angle that the desired line makes with the positive axis. (X-axis in case of X-y plane)

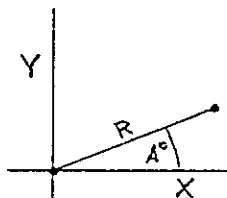
Example 19:1 Format:



In the above example R5.0 refers to the radial distance of the end point "B" from the start point "A".
"A45.0" refers to the angle that the desired line makes with the X axis. Note that the Polar definition is always incremental from the start (or previous) point. Both positive and negative angles can be specified.

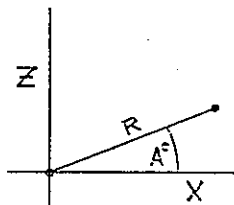
Polar Programming in other planes

XY Plane
(G17)

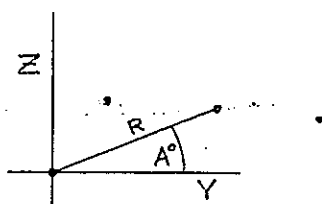


R=Radial distance
A=Specified angle

XZ Plane
(G18)



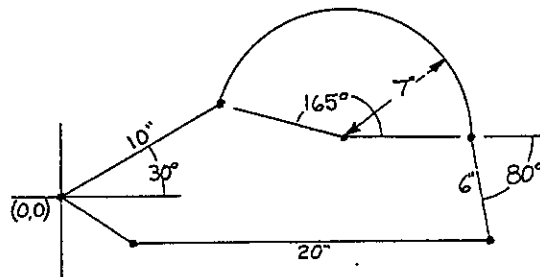
YZ Plane
(G19)



Since polar definitions for the circle and line take the previous position into account to figure out the incremental radial distance and angle, care must be taken if the points are approximations.

Example 19.2

```
X0 Y0
R10 A30
G2 R7 A165 A0
R6A-80
R20A-180
X0 Y0
```

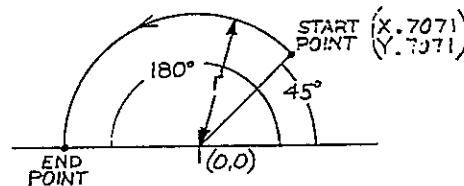


Circle case: Here the user has a choice of using polar definitions for both the center point and the end point or either the center or end point.

Eg. 19.3 G3 G17 R1 A450 A180.000

NOTE: POLAR CIRCLE DEFINITIONS MUST HAVE A "G2" OR A "G3" IN THE LINE.

In the Eg. 1 the CNC will move as follows:



Notice that the angles are always referred to from the 3 o'clock position in both the positive and negative direction.

Polar and Cartesian definitions can be mixed in a circle definition. The following commands will make the same circle.

```
G3 G17 R1 I0J0 A180
G3 G17 R1 A45 X-1Y0
G3 G17 R1 A45 A180
G3 G17 R1 I0 J0 X-1 Y0
```

Rotations/Scaling

Entire part programs can be rotated and/or scaled by changing Rotation parameter 93 and scaling parameters 84, 85, 86.

For scaling Parameter 84 scales X axis
 Parameter 85 scales Y axis
 Parameter 86 scales Z axis

Rotations/Scaling

Scaling

Entire part programs can be rotated and/or scaled by changing Rotation parameter 93 and scaling parameters 84, 85, 86.

For scaling Parameter 84 scales X axis
Parameter 85 scales Y axis
Parameter 86 scales Z axis

For 1:1 scale the scale factors should be "1". The scale factors can range from +99.9999. The program might not run beyond a certain range of scale factors due to tool diameter limitations. X, Y, and Z can be scaled independently, but circles both axis along the circle must have the same scale factor. After changing the scale factors the program must be re-cutter comped for the scale factors to take affect. The scale factors must be specified within the program to be in effect for that program. If a subroutine is to be called from a main program with different scale factors then the scale factors should be changed in the main program and the subroutine forced to be cutter-compensated by using a M6XX (where XX is the subroutine number) command before calling the subroutine. On power up the scale factors are forced to be "1". On running a program the scale factors are forced to a "1".

Rotations

Program parameter #93 is the Angle of rotation. Rotation is always about zero. This zero is the floating zero. If no floating zero is set then the rotation takes place about grid zero. On power up this parameter is zeroed out by the CNC. On running a program this parameter is zeroed out by the CNC. To achieve rotation the parameter must be changed within the parts program.

NOTE: Rotation only takes place in X-Y plane

Example 19.4 G393 P4.5
or G393 A45.0

will result in the program following the above instruction to be rotated by 45 degrees in the CCW direction. A negative rotation will result in CW rotation.

NOTE: When using rotation ALL lines in the program must contain both an X and Y dimension. If an X or Y dimension is omitted only the specified axis will be rotated. Autoroutines cannot be rotated.

USE OF PARAMETRIC SUBROUTINES

The toroid cavity mill subroutine can be used to generate the following shapes.

1. Toroidal section concave.
2. Toroidal section convex.
3. Cylindrical section concave.
4. Cylindrical section convex.
5. Conical section concave.
6. Conical section convex.

The various shapes are controlled by the following twelve parameters.

1. P18=Radius of the Toroid. If P18=0, the shape turns from a donut to a cylinder or cone.
2. P2=Length of cylinder or cone. (P18 must = 0).
3. P3=Radius of End 1 cross section.
4. P17=Radius of End 2 cross section.
NOTE: If P3=P17 cavity will be a cylinder. If P3≠P17 cavity will be a cone.
5. P15=End angle of cross section. The 9 o'clock position = 180 degrees, and the 3 o'clock position = 0 degrees, 360 degrees. If the end angle is between 180 degrees and 360 degrees, the cavity will be concave, between 0 degrees and 180 degrees the cavity will be convex.
6. P16=Start angle of cross section. The 9 o'clock position = 180 degrees and the 3 o'clock position = 0 degrees. If the start angle is between 180 degrees and 360 degrees the cavity will be concave, between 180 degrees and 0 degrees it will be convex.
NOTE: In the convex case the start angle must be greater than the end angle. In the concave case the start angle must be less than the end angle.
7. P10=Start angle of toroid. If P18=0 then P10 should =0. The 3 o'clock position = 0 degrees, the 6 o'clock position = 270 degrees.
8. P14=End angle of toroid. If P18=0, then P14 should = 0. The 3 o'clock position = 0 degrees, the 6 o'clock position = 270 degrees.
NOTE: P10, the start angle must always be greater than P14, the end angle.
9. P5=X and Y stepover increment. The number of thousandths X and Y move over each pass.
10. P7=Max Z increment. The maximum amount Z will increment down per pass.
11. P8=Thickness of block. This is the thickness of the block from the center of the cross section to the top of the piece. If the thickness of the block is less than the radius of the cross section, the cavity will have a flat bottom for concave, and a flat top for convex shapes.
12. P70=Tool Radius.

Parameter P93 part rotation can be used in conjunction with the cylinder and conical cavities to rotate them to any desired angle.

To run these cavities, load the toroid tape, delete any unwanted subroutines, write a main program that fills in the parameters and positions the machine to the start of the cavity and call subroutine 50. The starting point of the cavity is X0 Y0. The following 3 programs will demonstrate the 3 major variations of the Toroid Cavity.

Program 1 - EIA - 120 degrees toroid, 1"=Radius, .5"=Cross Radius, Concave, Z=0 Top of part.

```

1001 G100 G90 G80 G70 G60 G40 G30 G17
      G98
      M3
      G300 P.1      R-plane.
      G318 P1       Radius Toroid.
      G302 P0       Doesn't apply.
      G303 P.5      Cross section radius.
      G317 P.5      Cross section radius.
      G315 P360     End angle cross section.
      G101         Tool #1
      G316 P180     Start angle cross section
      G310 P120     Start angle toroid.
      G314 P0       End angle toroid.
      G305 P.1      X and Y stepover increments.
      G307 P.2      Max Z increment.
      G308 P.5      Block thickness.
      G370 P.125    Tool Radius.
      G0 X0 Y0      X-Y start of cavity.
      G1 F25 G850   Set feedrate do cavity.
      GOG100 Z0
      G0 X0 Y0
      M5

```

Program 2 - EIA - Cylinder, Length 2", Rotate 120 concave, cross section .5

```

      G100 G90 G80 G70 G60 G40 G30 G17
      G98
      M3
      G393 A120     Rotate 120 degrees.
      G0 G101 X1 Y0 Start point cylinder.
      G302 P2       Length of Cylinder.
      G300 P.1      R-plane
      G318 P0       Not used.
      G303 P.5      Cross section Radius.
      G317 P.5      Cross section Radius.
      G315 P360     End angle
      G316 P180     Start angle.
      G310 P0       Not used.
      G314 P0       Not used.
      G305 P.1      X Y increments.
      G307 P.2      Z increment.
      G308 P.5      Block thickness.
      G370 P.125    Tool radius.
      G1 F25 G850   Set feedrate: do cavity.
      G0 G100 Z0
      X0 Y0
      M5

```

Program 3 - EIA - Cone, Lenght 2", Rotate 180 degrees, End 1 R=.5,
End 2 R=.2, Convex

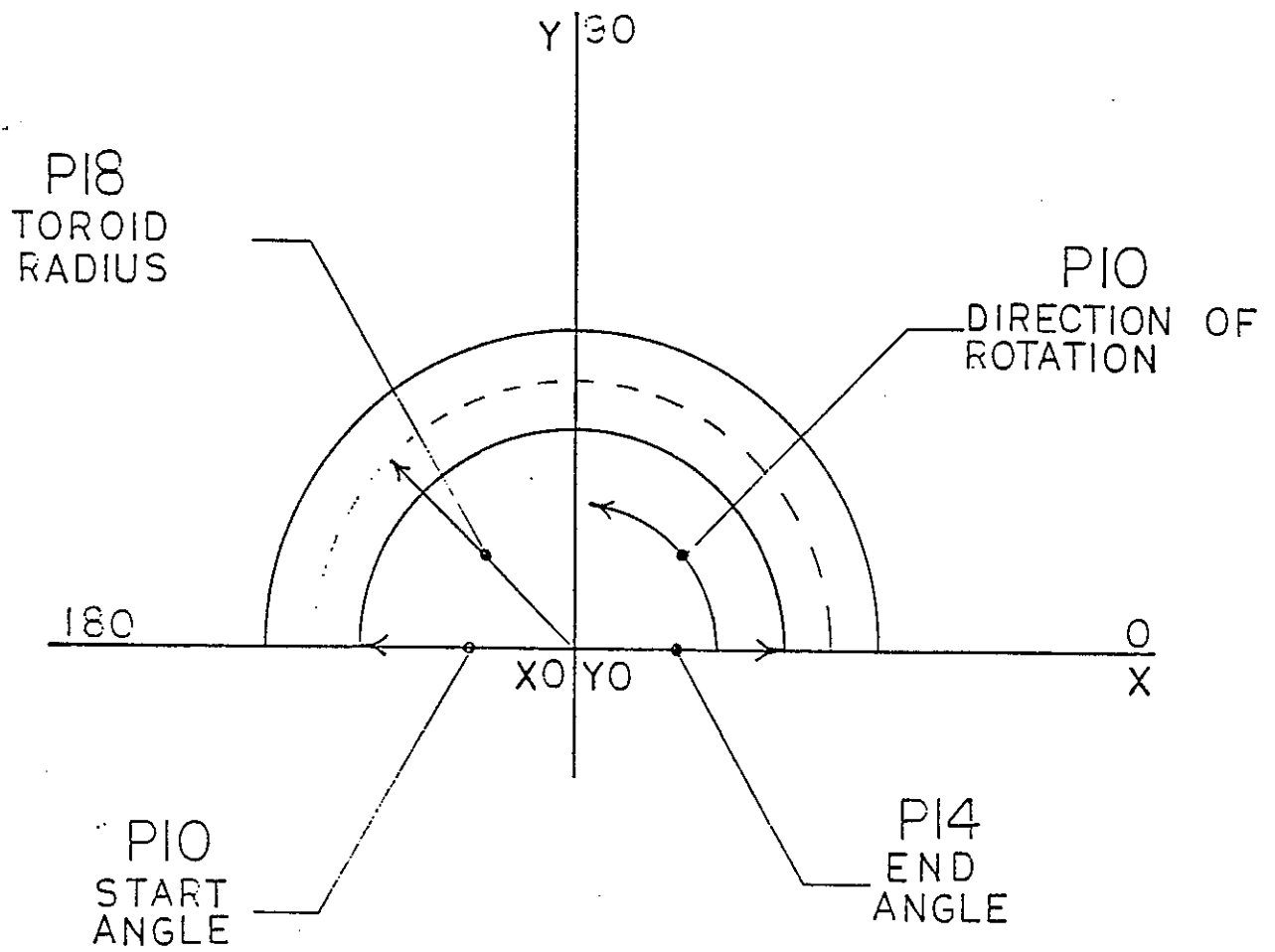
```
G100 G90 G80 G70 G60 G40 G30 G17
G98
M3
GO G101 X1 Y0
G302 P2
G300 P.1
G318 P0
G303 P.5 Radius End 1
G317 P.2 Radius End 2
G315 P0
G314 P0
G305 P.1
G307 P.2
G308 P.5
G370 P.125
G393 A180
G1 F25 G850
GO G100 Z0
X0 Y0
M5
```

Program 4 - Same as program 2, but written in Help Mode.

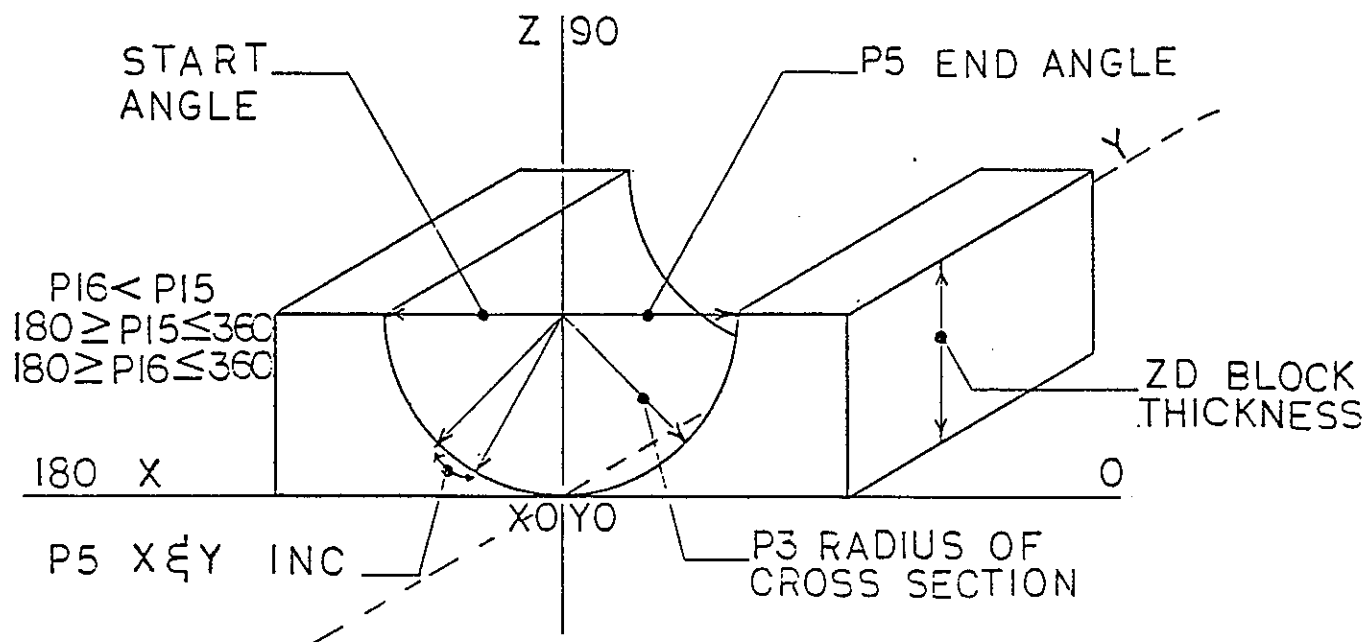
<u>Event-#-</u>	<u>Event-Type-</u>	<u>Data-</u>
1	Init. Program	Abs. Inch X1, Y1, Z1, A120
1	Tool change	X0, Y0, T1, SP, CW
1	Move	F999, X1, Y0
2	Modify Parameter	Load P03 P.5
3	"	" P00 P.1
4	"	" P5 P.1
5	"	" P2 P2
6	"	" P17 P.5
7	"	" P15 P360
8	"	" P16 P180
9	"	" P10 P0
10	"	" P14 P0
11	"	" P18 P0
12	"	" P7 P.2
13	"	" P8 P.5
14	"	" P70 P.125
15	Misc.	F25
16	Call Sub.	50 Loop 0
17	Tool Change	X0 Y0 T0

Note: Events 2, 3, 4, won't display because these parameters are normally part of Autoroutines.

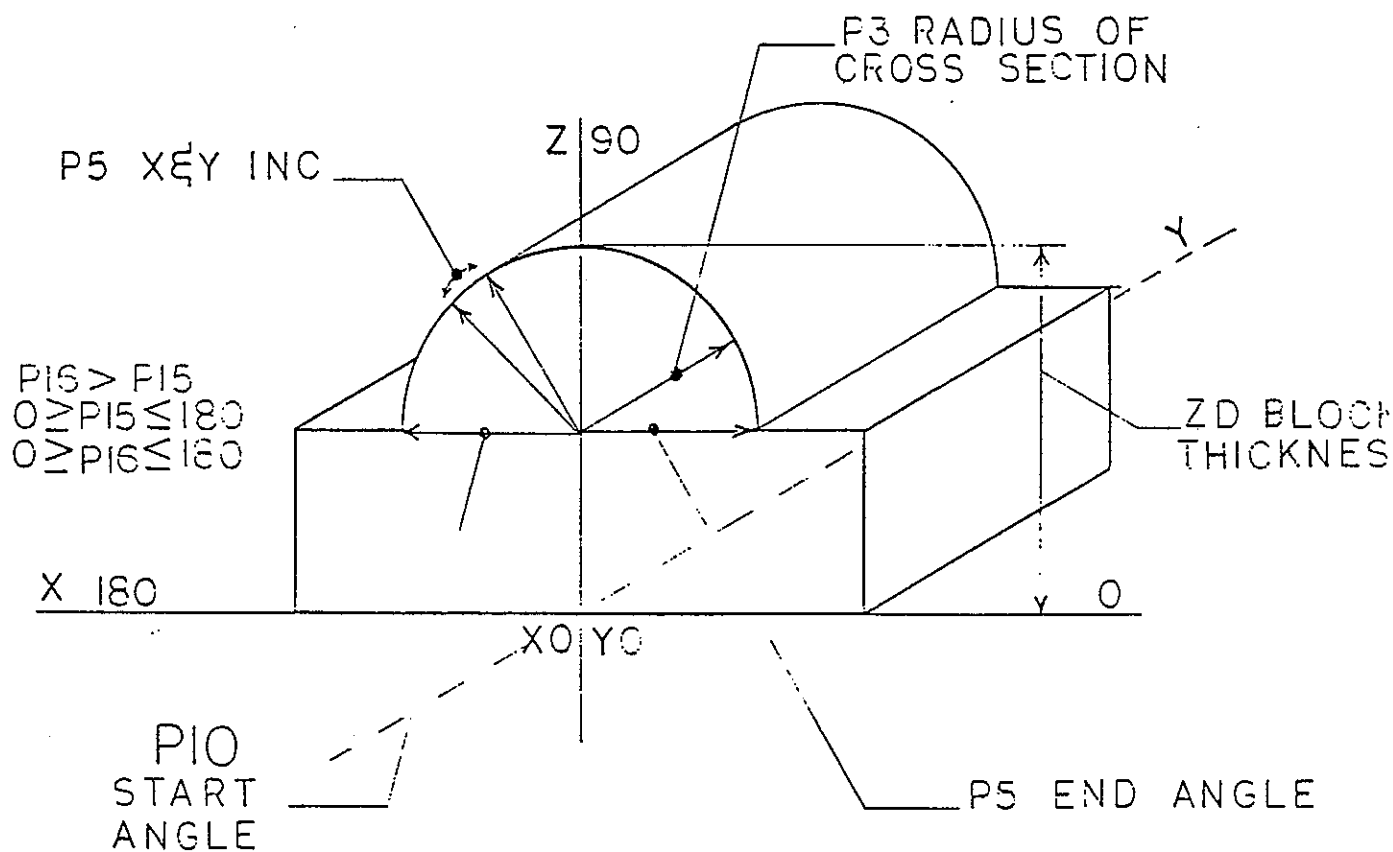
CROSS SECTION



TOROID CAVITY



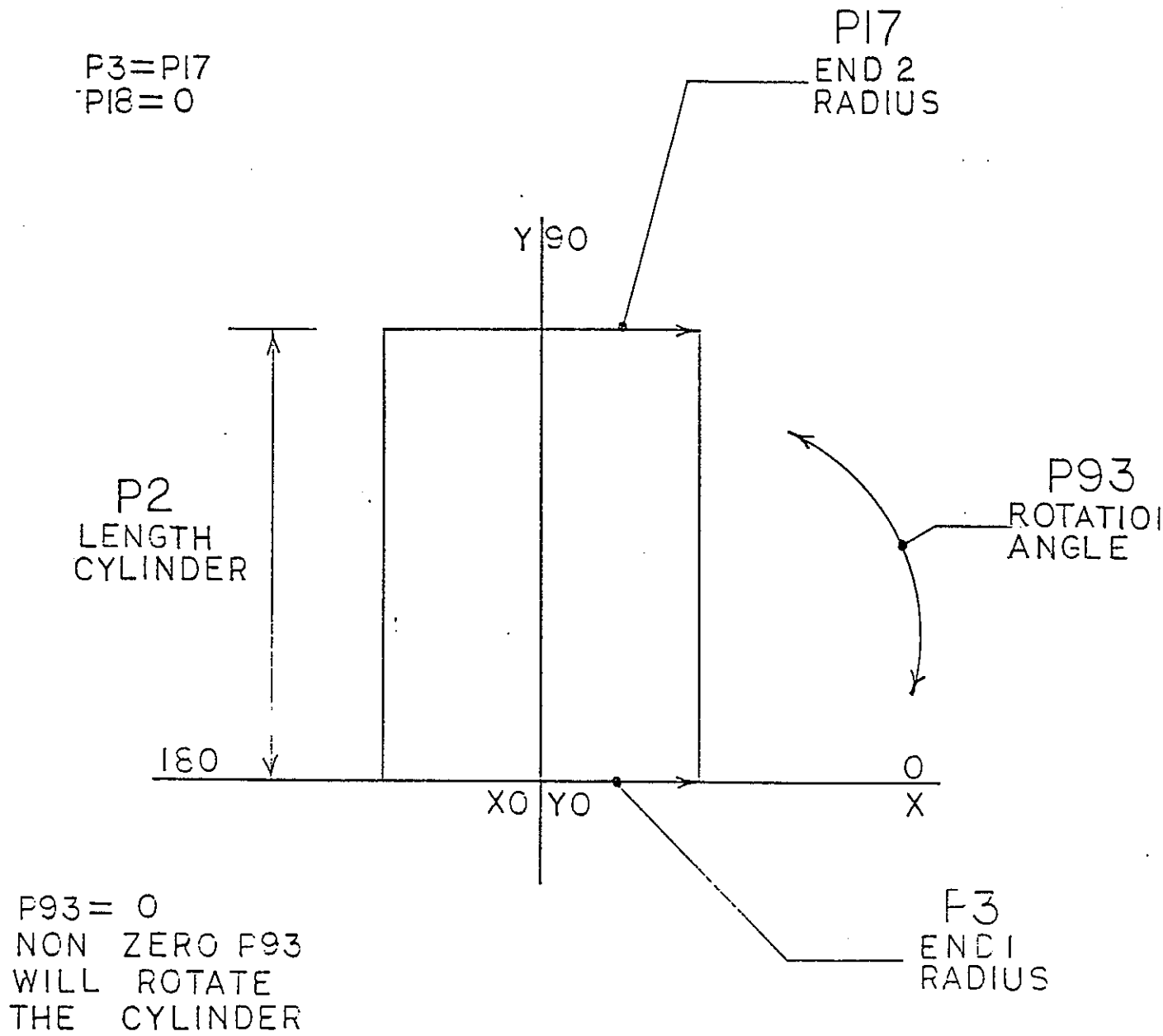
CONCAVE



CONVEX

TOROID CAVITY

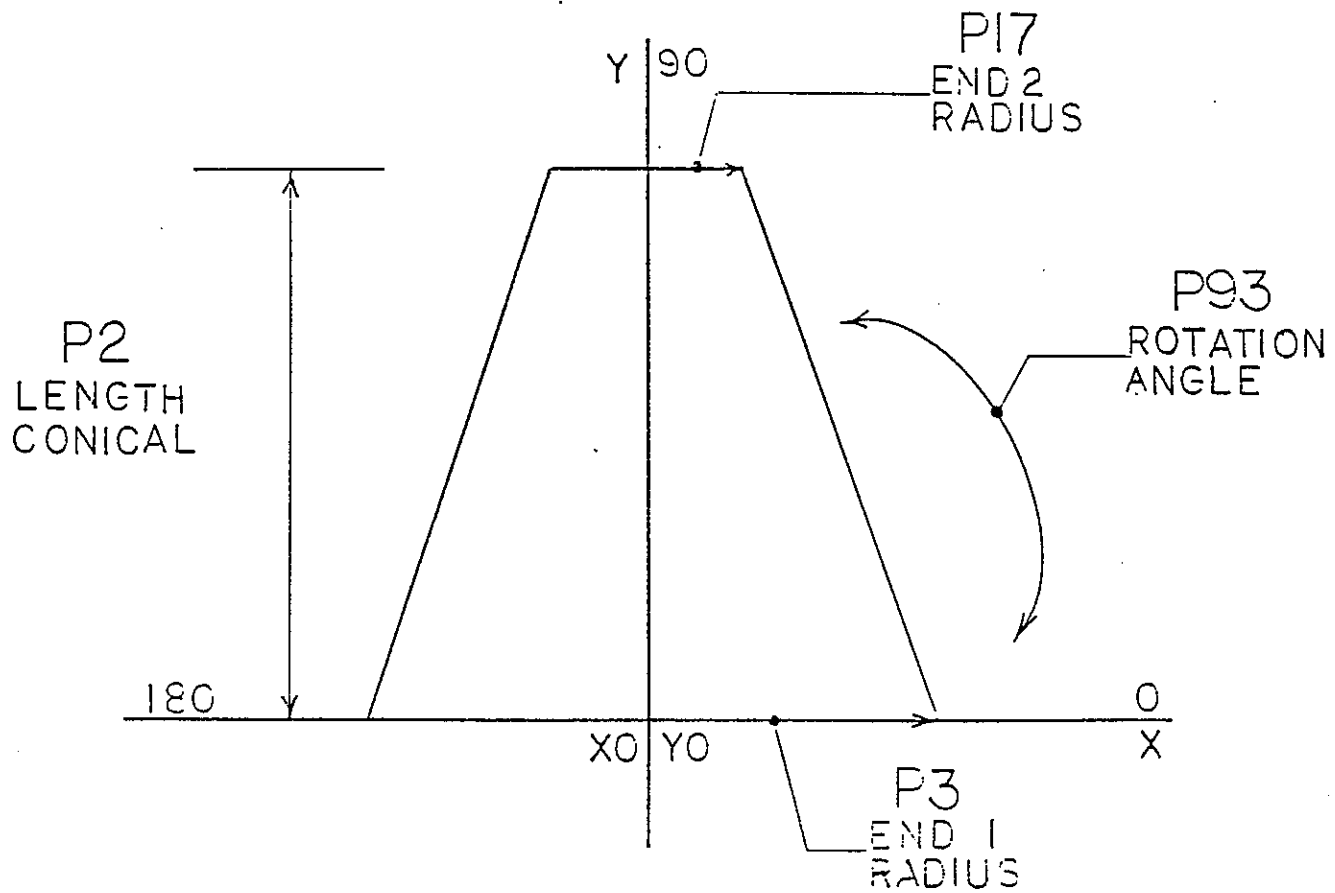
CROSS SECTION



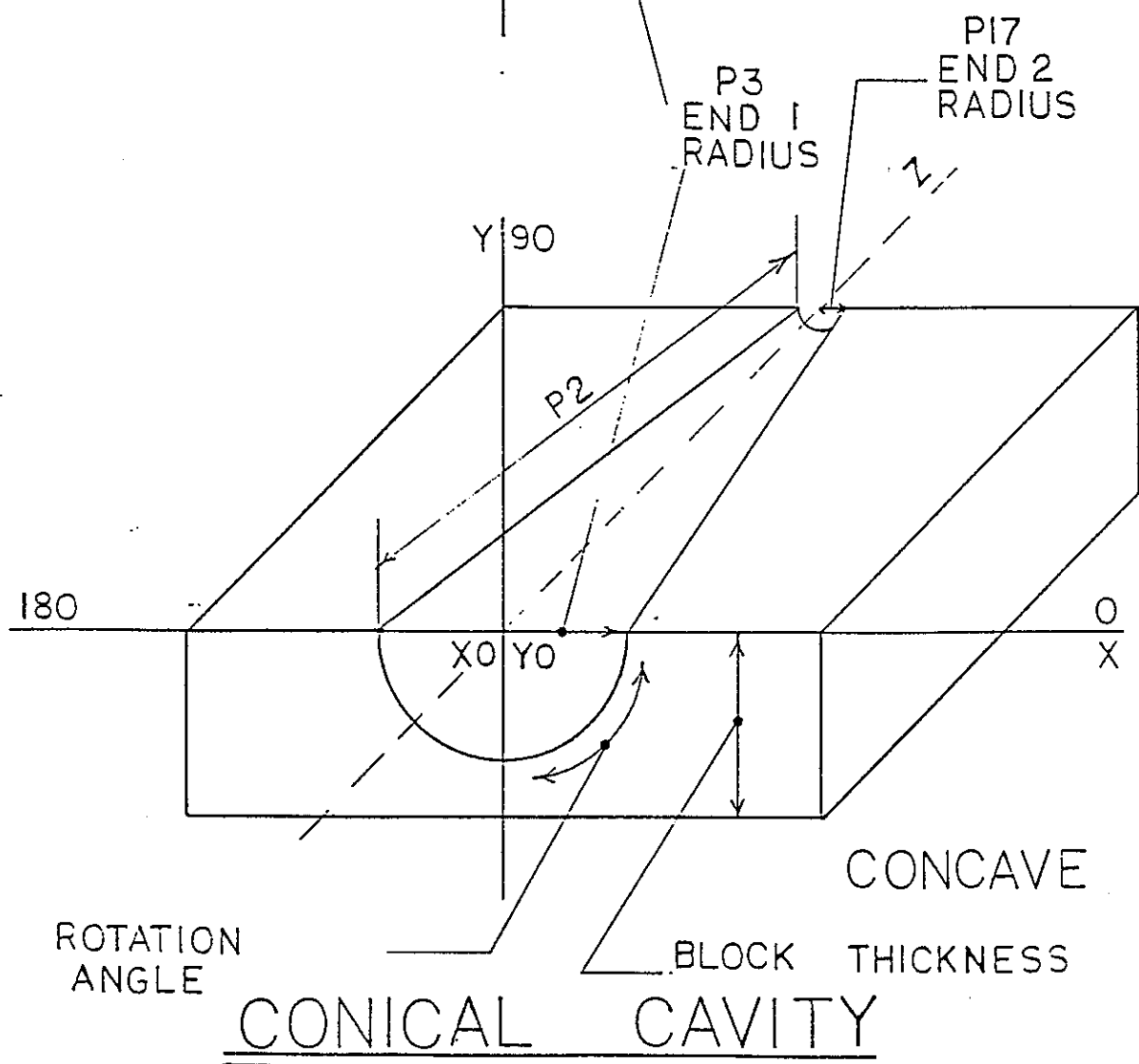
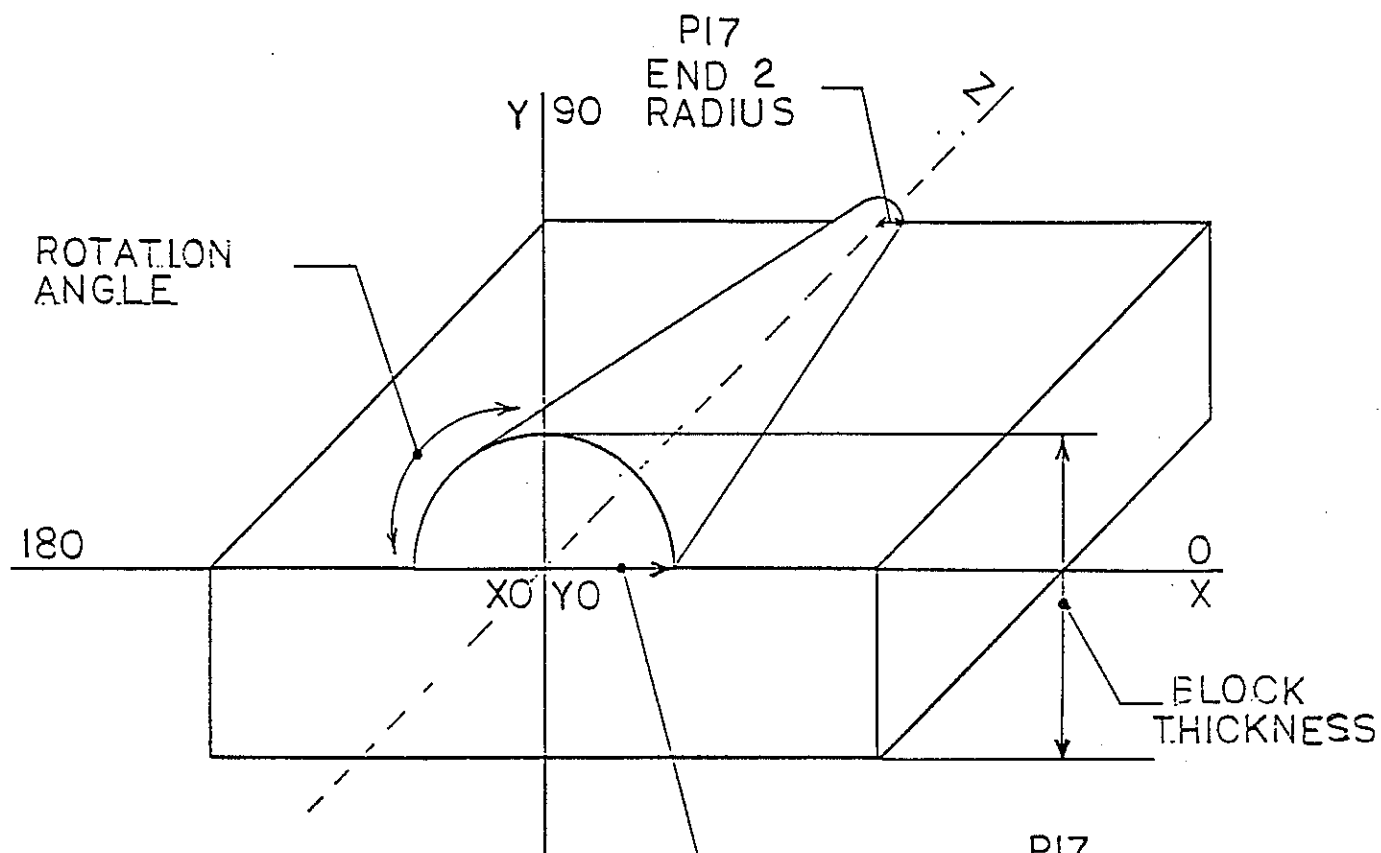
CYLINDER CAVITY

CROSS SECTION

$P3 \neq P17$
 $P18 = 0$



CONICAL CAVITY



TRIANGLE CAVITY

The triangle cavity subroutine will generate a triangular shaped cavity or boss. The following parameters are needed to specify the cavity,

1. P3=The distance from the center of the triangle to the corners at the top.
NOTE: The triangle must be equilateral.
2. P7=Z - Increment.
3. P8=Final Z depth.
4. P11=The distance from the center of the triangle to the corners at the bottom.
NOTE: The combination of P11 and P3 determines the slope of the walls. If P11 is less than P3, the part will be a cavity. If P11 is greater than P3, the part will be a boss.
5. P10=The angle from the center of the triangle to the 1st vertex. The 3 o'clock position is 0 degrees.
6. P70=Tool Radius.

To run the cavity or boss load the triangle subroutine section of tape, delete any unwanted subroutines and write a main program that fills in the above parameters. The following two programs will cut a triangular cavity and boss.

```
PROGRAM 1 - TRIANGULAR CAVITY,.4 deep,.4 bottom radius,1 top radius
1001 G100 G90 G80 G70 G60 G40 G30 G98 M3
      G101 G001 F25 - Tool #, Feedrate
          G300 P.1 - R-plane
          G303 P1 - Size of top of triangle
          G311 P.4 - Size of bottom of triangle
          G307p.1 - Z inc.
          G308p-.4- Final Z depth
          G310p90 - Start angle of first vertex
          G370p.2 - Tool radius
          G851 - Subroutine call
```

```
PROGRAM 2 - TRIANGLE BOSS
2001 G100 G90 G80 G70 G60 G40 G30 G98 M3
      G101 G001 F25
          G300 P.1
          G303 P.5
          G311 P1.2
          G307p.1
          G308p-.4
          G310p45
          G370p.2
          G851
```

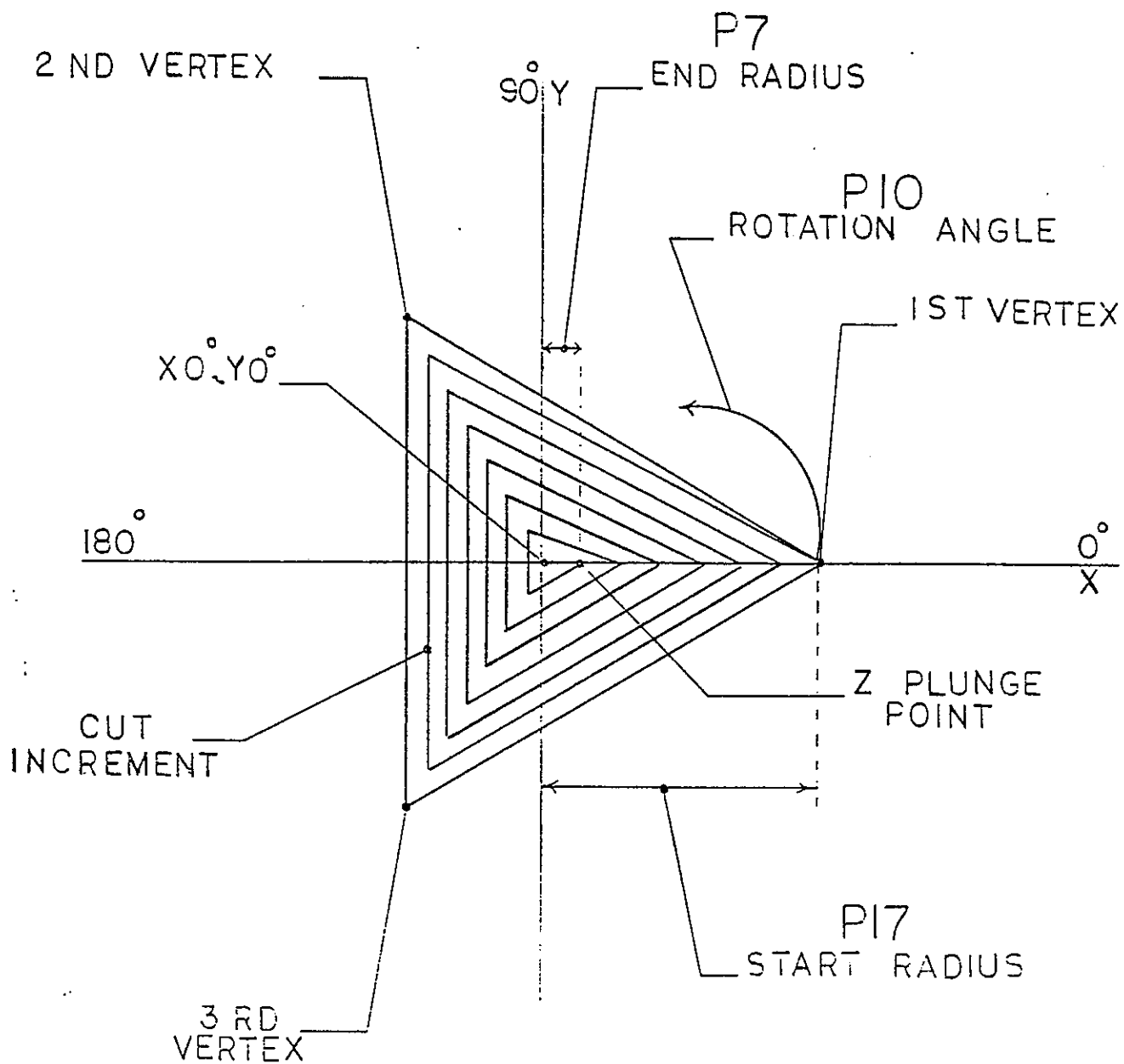
PROGRAM 3 - HELP

Event #	Event Type
1	Create Program
2	Tool Change #1
3	Modify Para.
4	"
5	"
6	"
7	"
8	"
9	"
10	Misc.
11	Call Sub.

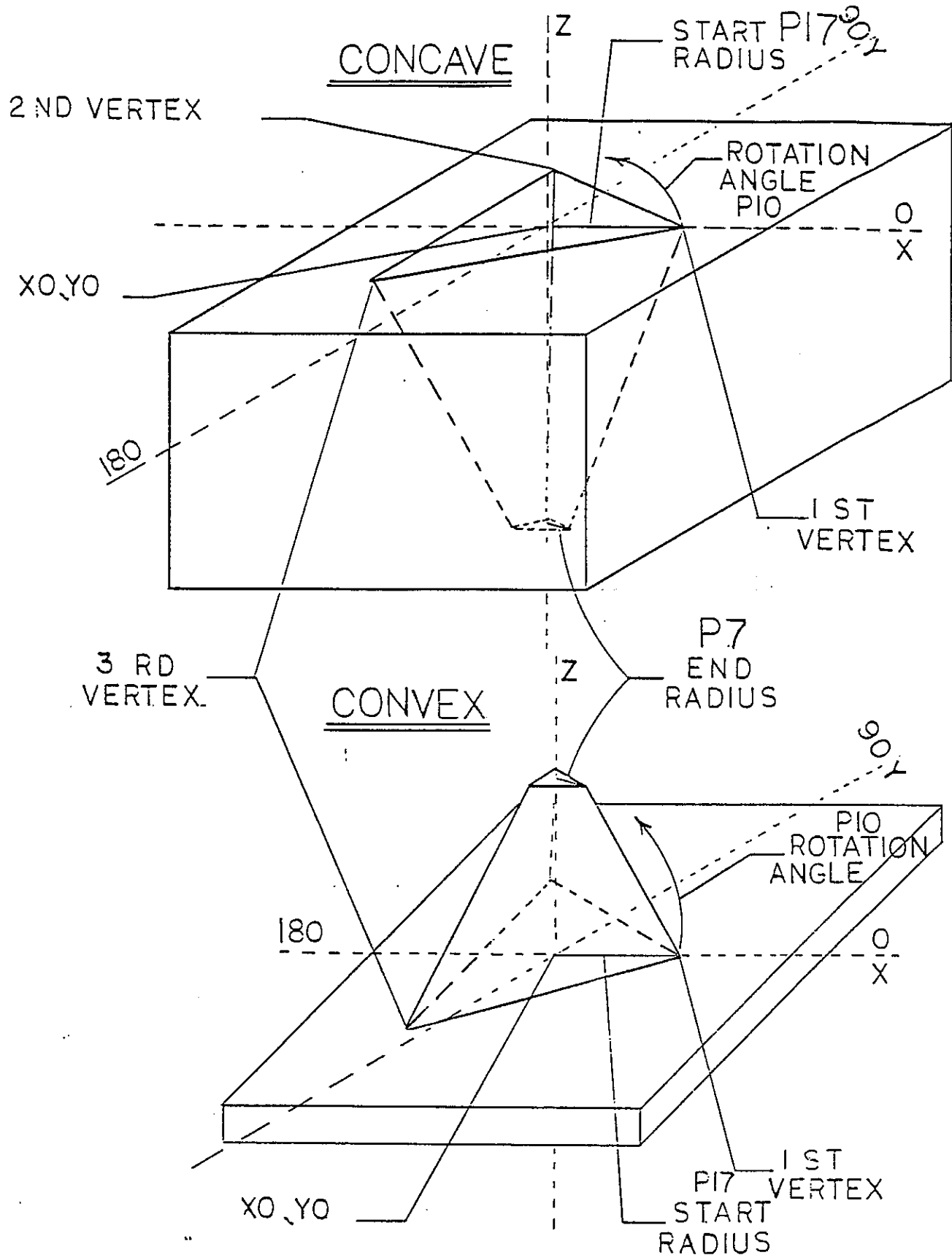
PROGRAM - 2 - TRIANGLE BOSS

Data
P#3, ABS, Inch, X=1, Y=1, Z=1, A0
X0, Y0, A0, CW
Load Para, N0, P.1
" N3, P.5
" N11, P1.2
" N7, P.1
" N8, P.4
" N10, P45
" N70, P.2
F25
Sub, #51 Loop 0

BOTTOM VIEW



TRIANGLE CAVITY



TRIANGLE CAVITY

SPHERICAL CAVITY

The spherical cavity subroutine will generate a spherical shaped cavity or boss. The following parameters are needed to specify the cavity.

1. P3=The radius of the sphere.
2. P5=The XY increment of the clearing passes.
3. P7=The maximum Z increment.
4. P70=The tool radius.
5. P78=Number of degrees of the sphere to cut in the XY plane. This number should always be negative.
6. P80=Number of degrees of the sphere to cut in the XZ plane. This number is always between 0 degrees and 90 degrees. The number is negative for a concave cavity and positive for a convex boss.
7. P0=Clearance Plane.

To run the cavity or boss load the sphere subroutine section of tape, delete any unwanted subroutines and write a main program that fills in the above parameters. The following two programs will cut a spherical cavity and boss.

PROGRAM 1 - Spherical Cavity, 360 degrees XY, 90 degrees XZ, 1 inch radius.

```
1001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G300 P,1          R Plane
      G101 G1 F25 G303 P1 Tool #1, Feedrate 25 , Radius Sphere=1
      G305 P,125        XY Increment
      G307 P,1          Maximum Z Increment.
      G370 P,125        Radius of Tool.
      G378 P-36,0       360 degrees of Arc X Y
      G380 P-90         90 degrees of Arc XZ
      G853             Run Subroutine
```

PROGRAM 2 - Spherical Boss, 270 degrees XY, 90 degrees XZ, .75 inch radius.

```
2001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G300 P,1
      G101 G1 F25 G303 P,75
      G305 P,1
      G307 P,09
      G378 P-270
      G380 P90
      G370 P,125
      G853
```

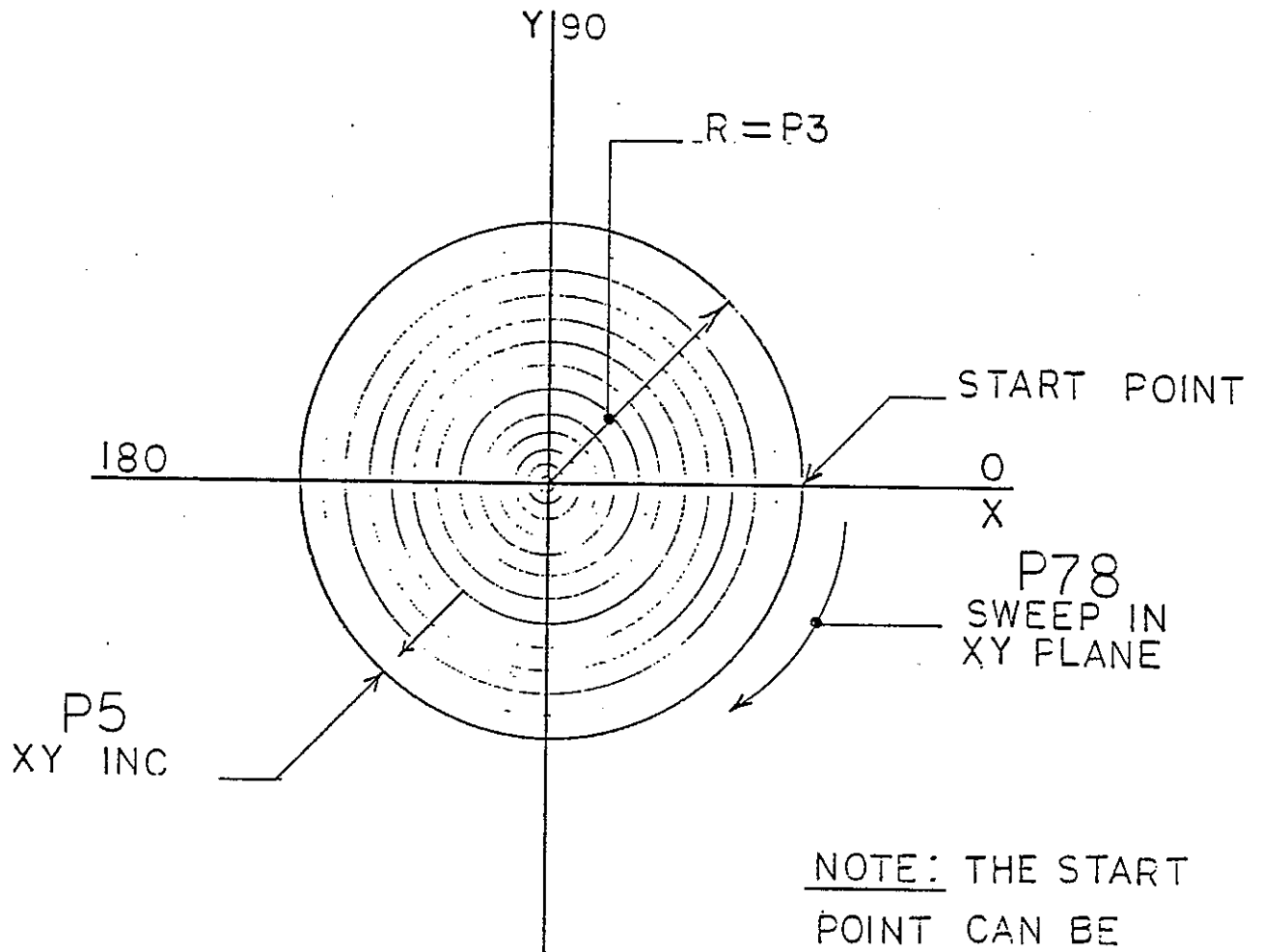
PROGRAM 3 - Spherical Boss, 180 degrees XY, 45 degrees XZ, .75 inch radius.

```
3001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G300 P,1
      G101 G1 F25 G303 P,75
      G305 P,1
      G307 P,09
      G378 P-180
      G380 P90
      G370 P0.125
      G853
```

PROGRAM 4 - Spherical Cavity, 260 degrees XY, 40 degrees XZ, 1 inch radius,
Help.

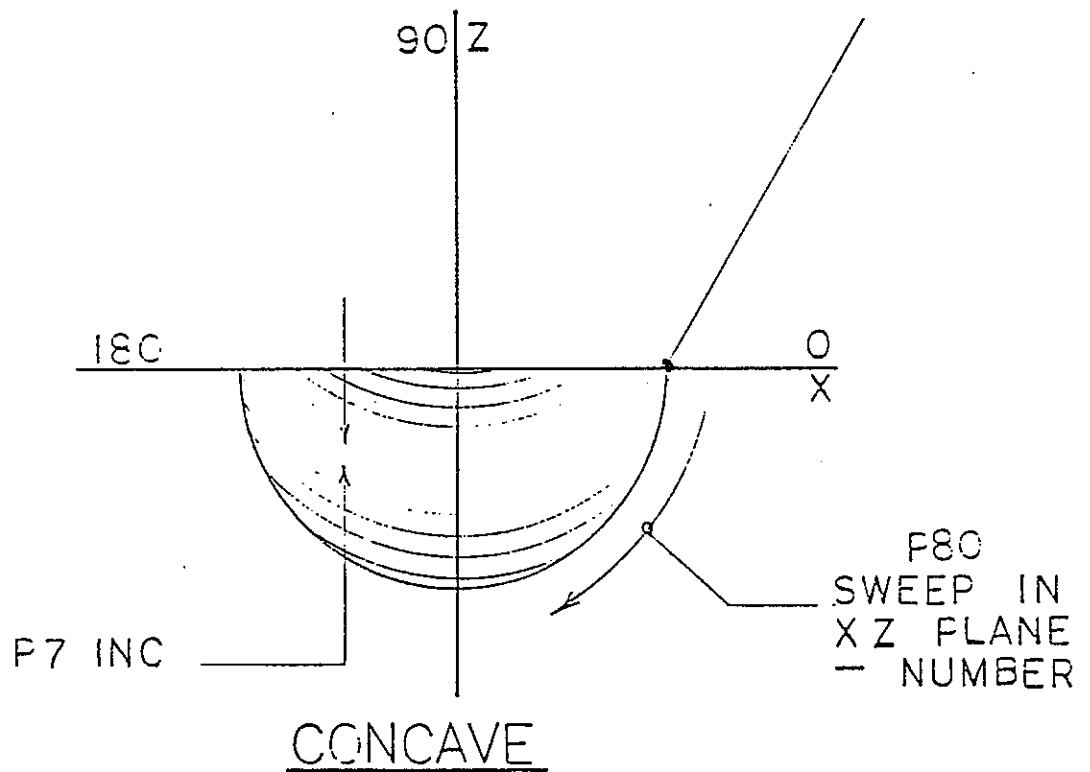
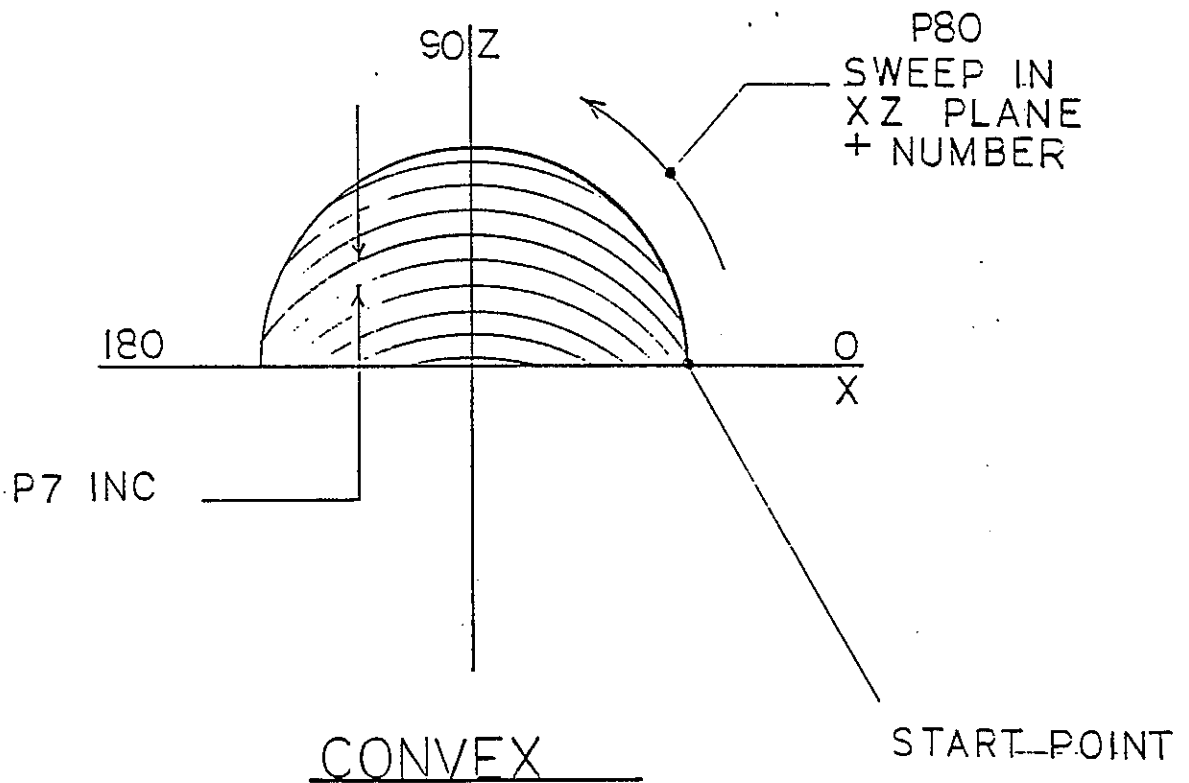
<u>Event #</u>	<u>Event Type</u>	<u>Data</u>
1	Create Program	P#4, ABS, Inch, X=1, Y=1, Z=1, A=0.
2	Tool Change #1	X0, Y0, A0, CW, T1
3	Modify Parameter	Load Parameter N0 P.1
4	"	" N3 P1
5	"	" N5 P.125
6	"	" N7 P.1
7	"	" N70 P.125
8	"	" N78 P-260
9	"	" N80 P-40
10	Move	F25 XoYo
11	Call Sub.	Sub, #53 Loop 0
12	Tool Change	XoYo Ao CW To

CROSS SECTION



NOTE: THE START
POINT CAN BE
ROTATED BY USING
STANDARD ROTATION
PARAMETER 93.

SPHERICAL CAVITY



SPHERICAL CAVITY

RECTANGULAR POCKET CLEAR WITH ROUND CORNERS

The rectangular pocket clear will clear any rectangular shaped pocket with round or square corners. The following parameters are needed to generate the pocket.

1. P0 Clearance Plane,
2. P1 X Pocket length,
3. P2 Y Pocket length,
4. P3 Corner radius,
5. P5 XY stepover increments,
6. P6 1st Z increment,
7. P7 Next Z increment,
8. P8 Z final depth,
9. P70 Tool radius,

NOTE: This subroutine can be rotated or scaled using parameters 93, 84, 85, 86. If P1=2 P3 the subroutine clears out a race track shaped pocket along Y axis or along X axis if P2=2•P3.

PROGRAM 1 - Rectangular Pocket X=3, Y=2, Corner R=.5, Depth=-.5, Tool=.125,

```
1001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G1 F20 G101 G300 P,1      Feedrate, Tool 1, R-Plane
      G301 P3                  X Pocket length
      G302 P2                  Y Pocket width
      G303 P,5                  Corner radius
      G305 P,1                  XY stepover increment
      G306 P-,1                 1st Z increment
      G307 P,2                  Next Z increment
      G308 P-,5                 Final Z depth
      G370 p.125
      G855                      Subroutine call cut pocket
```

PROGRAM 2 - Race track, at 45 degrees, X=3, End Radius=1, Tool .125, Depth=-.5,

```
2001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G1 G101 G300 P,1
      G301 P3
      G302 P2 P2=2•P3
      G303 P1
      G305 P,1
      G306 P-,1
      G307 P,2
      G308 P-,5
      G393 P4,5 Rotate 45 degrees
      G370 p.125
      G855 Cut pocket
```

RECTANGULAR CAVITY WITH DRAFT

The rectangular cavity subroutine will generate a rectangular shaped cavity or boss, with either round or square corners. The following parameters are needed to specify the cavity,

1. P0 - Clearance Plane.
2. P1 - X length of pocket.
3. P2 - Y width of pocket.
4. P3 - Radius of corners.
5. P5 - X stepover increment.
6. P6 - 1st Z increment.
7. P7 - Next Z increment.
8. P8 - Final Z depth
9. P70 - Tool Radius.
10. P4 - Slope of sides.
11. P9 - 0=Mode 0 or 1=Mode 1 in Mode 0 as the draft increases or decreases the radius in the corners will stay the same. In Mode 1 the corner radius will increase or decrease with the draft. In Mode 1 if $P1=2 \cdot P3$ and $P2=2 \cdot P3$ the subroutine will cut either conical cavity or boss. In Mode 0 if $P3=0$ the boss or cavity will have square corners.
NOTE: This subroutine can be rotated using Parameter 93 or scaled using Parameters 84, 85, 86. Z zero should be the top of the part.

PROGRAM 1 - Draft=45 degrees, Corner Radius=.5, ToolRadius=.125, Cavity X=3", Y=2", Depth=.5, Mode=0 constant corner radius.

```
1001 G100 G90 G60 G80 G70 G40 G30 G98 M3
1002 G1 F20 G101 G300 P,1 - Feedrate, Tool 1, R-Plane.
1003 G301 P3          X dimension of pocket
      G302 P2          Y dimension of pocket
      G303 P,.5        Corner Radius.
      G304 P-45,0      Draft angle - makes cavity.
      G306 P0          1st inc. if P6=P8. Z will inc. to bottom along slanted wall.
      G307 P,.05       Next Z increment
      G308 P-,.5       Z depth
      G309 P0          Mode=0
      G370 P,.125      Tool Radius
      G854             Call sub. and cut cavity.
```

PROGRAM 2 - Draft=30 degrees, Corner Radius =.5, ToolRadius=.125, Boss X=3, Y=2, Depth=.5, Mode=1 changing corner radius.

```
1001 G100 G90 G60 G80 G70 G40 G30 G98 M3
      G1 F20 G101 G300 P,1
      G301 P3
      G302 P2
      G303 P,.5
      G304 P30          Draft angle + makes boss,
      G306 P0          1st inc.
      G307 P,.05
      G308 P-,.5
      G309 P1          Mode=1
      G370 P,.125
      G854
```

PROGRAM 3 - Cone, Draft 45, Radius=.5, Tool=.125, Boss, Mode=1, Depth .5.
1001 G100 G90 G60 G80 G70 G40 G30 G98 M3

G1 F20 G101 G300 P.1

G301 P1

G302 P1

G303 P.5

G304 P45

G306 P0 1st inc.

G307 P.05

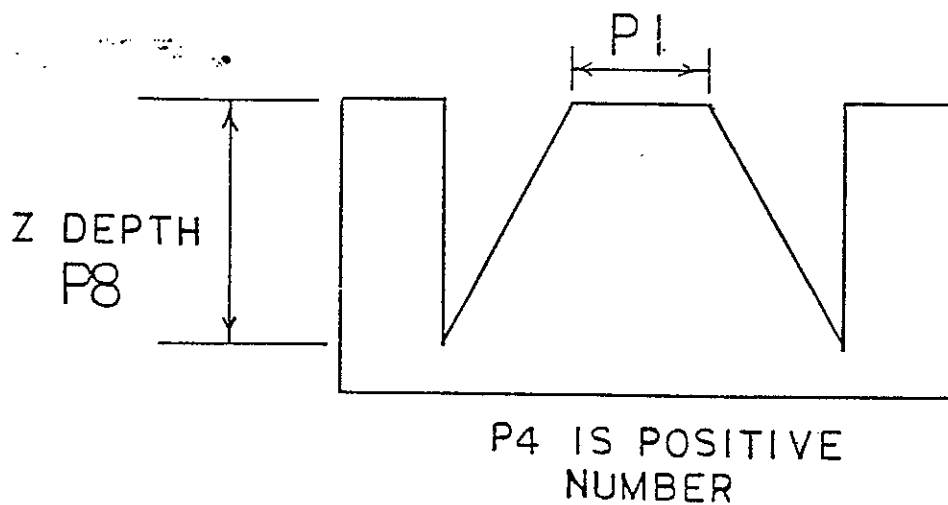
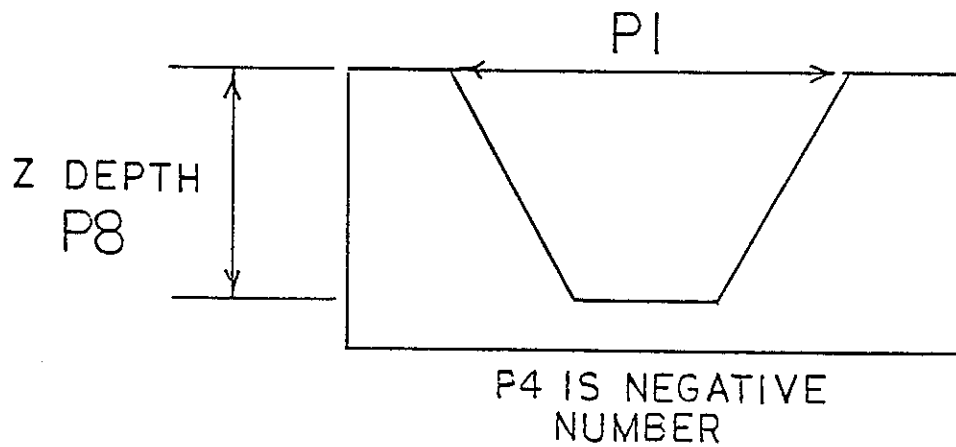
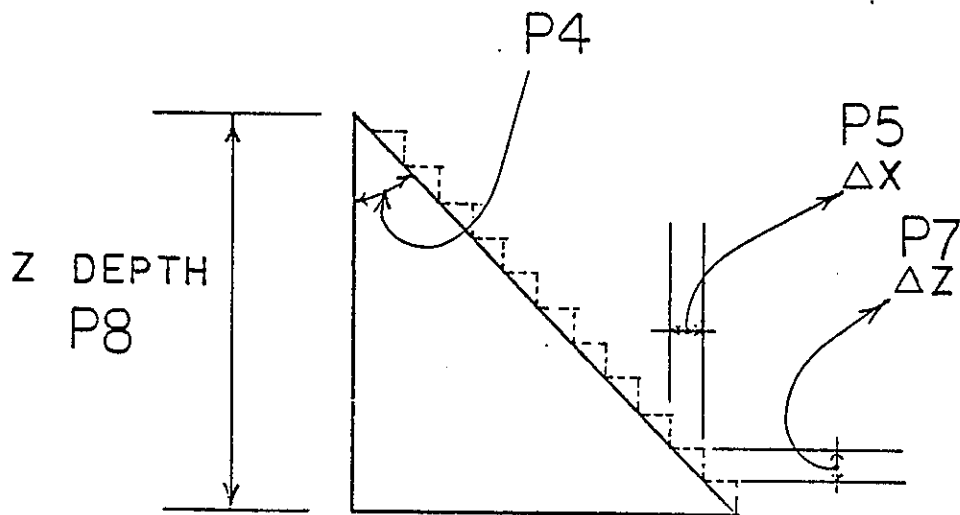
G308 P-.5

G309 P1

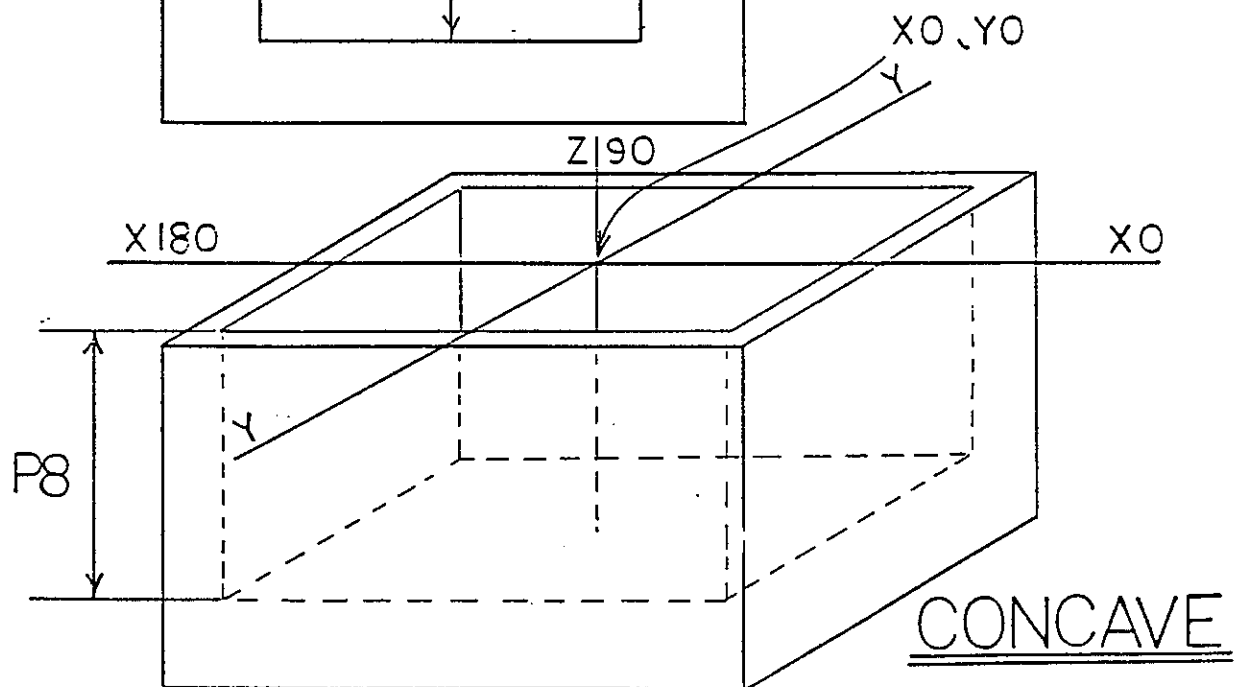
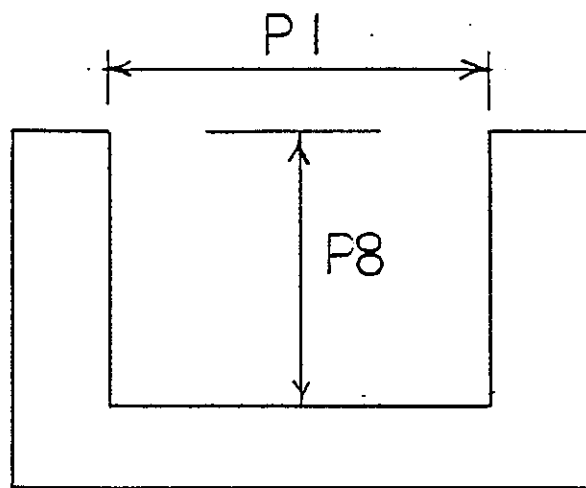
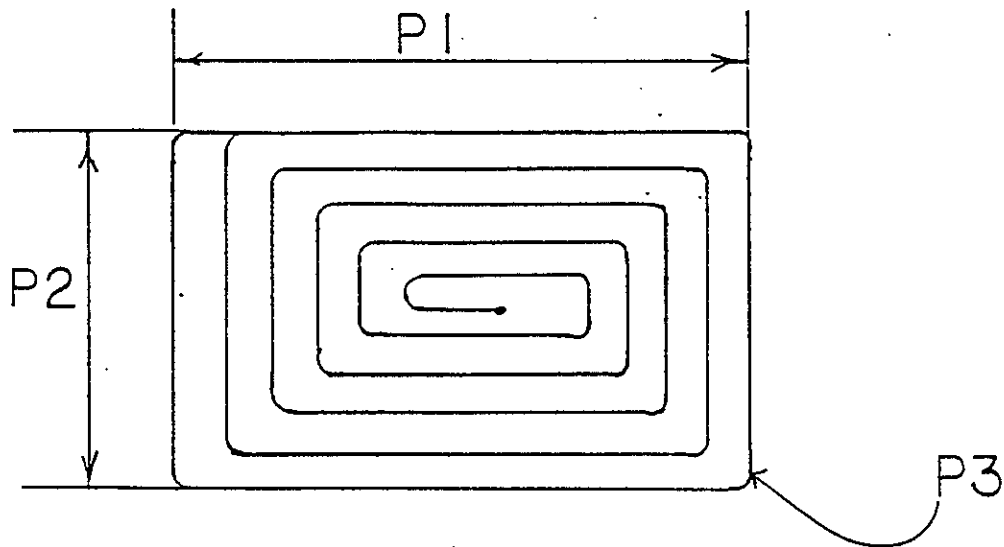
G370 P.125

G854

RECTANGULAR CAVITY



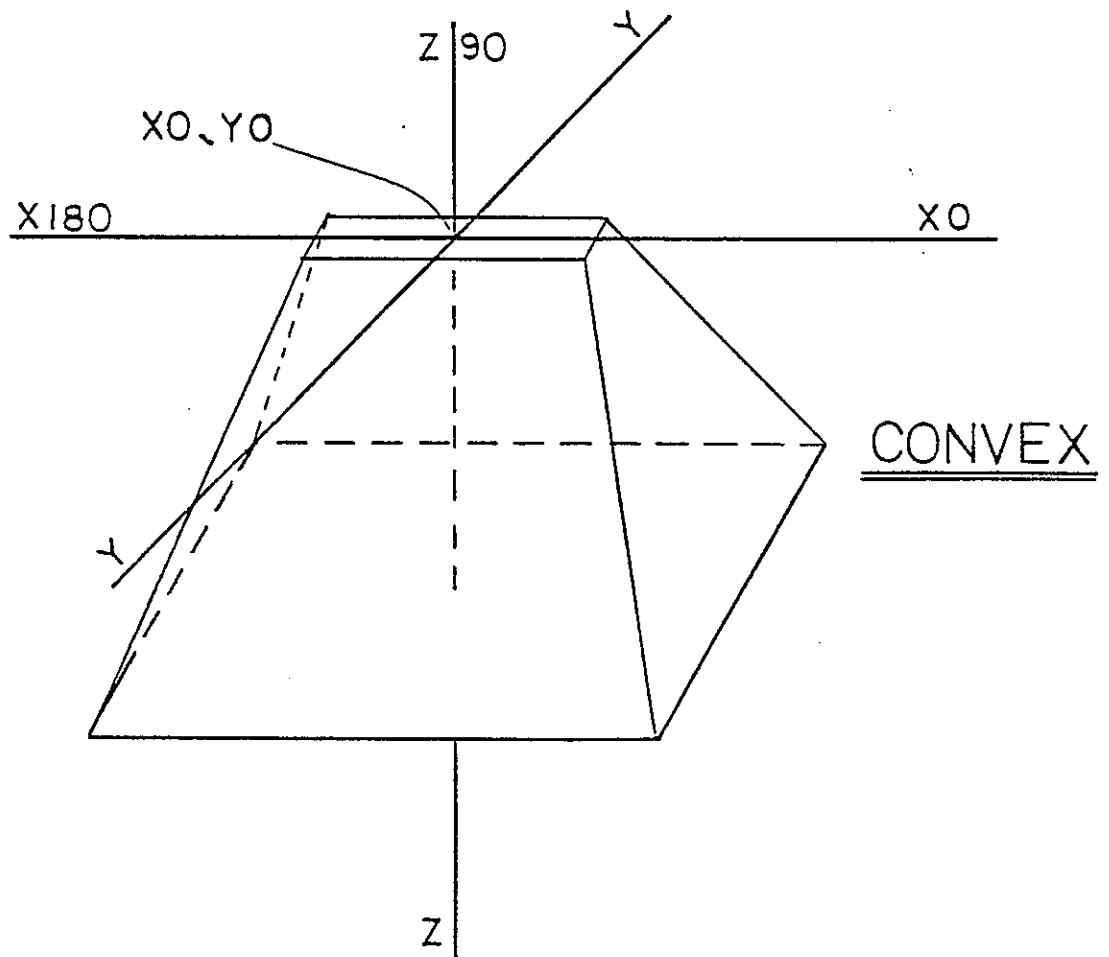
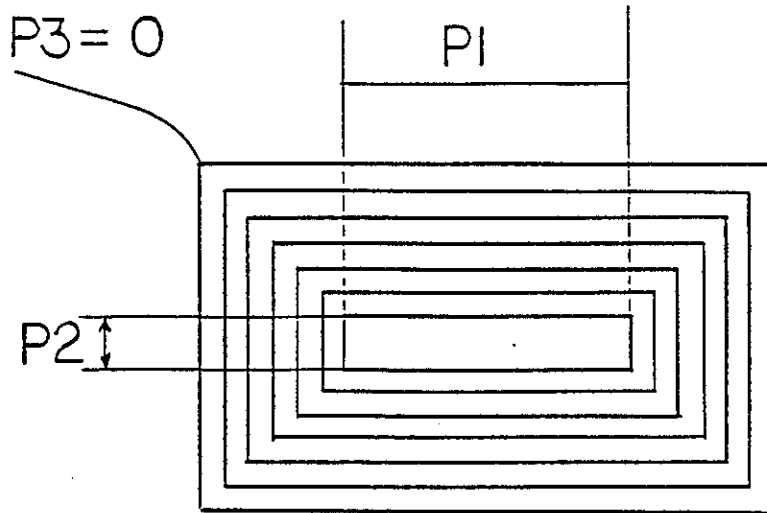
RECTANGULAR CAVITY CLEAR



P4 IS
NEGATIVE

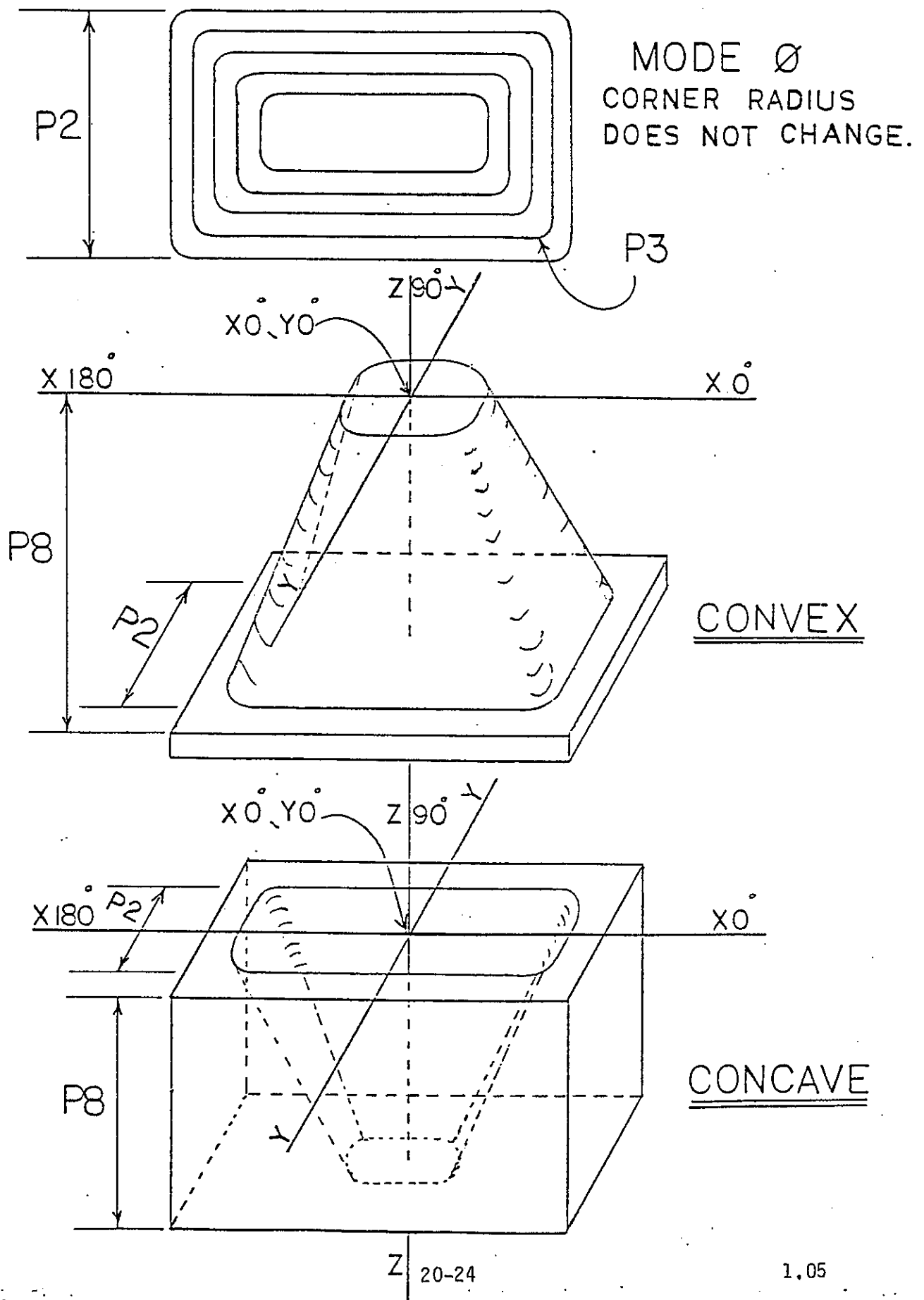
Z

RECTANGULAR CAVITY MODE ZERO

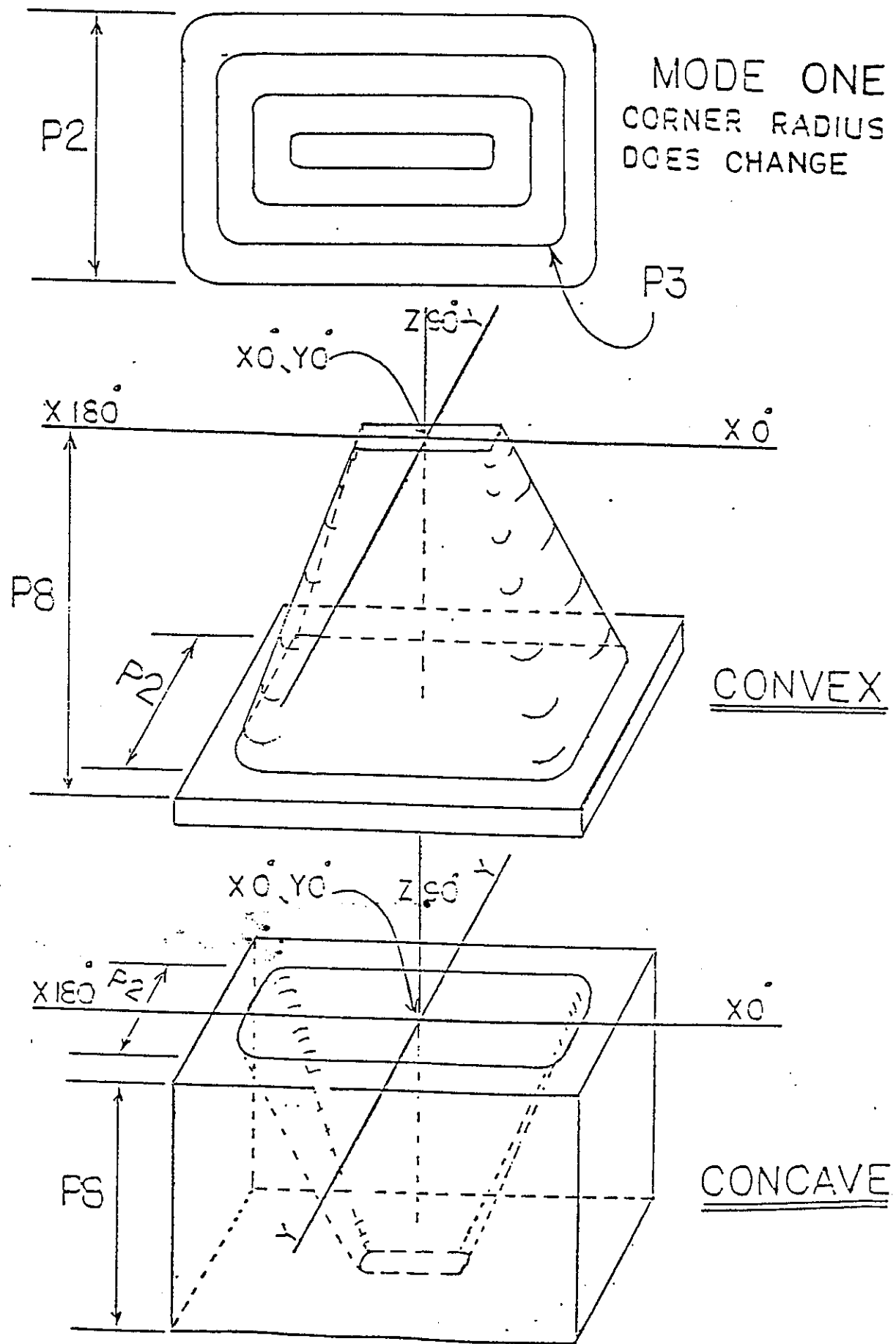


P4 IS POSITIVE

RECTANGULAR CAVITY WITH DRAFT



RECTANGULAR CAVITY WITH DRAFT



USE OF THE ALPHABET TAPE

The alphabet tape is a series of subroutines which will cut the alphabet in approximately 1" high letters. Subroutines 60 thru 69 are numbers 0 thru 9. Subroutine 60 is the number 0 and subroutine 69 is the number 9. Subroutines 70 thru 95 are the alphabet. 70 is the letter A and 95 is the letter Z. Subroutines 96 thru 99 are the characters -, +, ., and ,. These letters and numbers can be scaled or rotated to achieve any desired height or angular displacement. The subroutine can be called and run from either the help or standard EIA modes. The following example shows how to write these two messages.

LETTER OR NUMBER AND ASSOCIATED SUBROUTINE NUMBER

<u>Character</u>	<u>Subroutine No.</u>	<u>Character</u>	<u>Subroutine No.</u>
0	60	L	81
1	61	M	82
2	62	N	83
3	63	O	84
4	64	P	85
5	65	Q	86
6	66	R	87
7	67	S	88
8	68	T	89
9	69	U	90
A	70	V	91
B	71	W	92
C	72	X	93
D	73	Y	94
E	74	Z	95
F	75	-	96
G	76	+	97
H	77	.	98
I	78	,	99
J	79		
K	80		

The above characters have a predetermined distance established between characters. If, however, it is desirable to increase this distance, write a subroutine 01 followed by the additional spacing dimension and call up this subroutine between each character subroutine.

Help - Mode Program 1

Screens Initialize

Comp. Sub.

Move

Cut-Mill

Misc.

Subroutine

Subroutine

Subroutine

Move

Cut-Mill

Subroutine

Subroutine

Subroutine

Subroutine

Data

ABS Inch

X=1, Y=1, Z=1, A=0.

All

X - 1.6, Y1.5.

F50, Z 1st 0, Z inc. 1, R.1, Z depth 0.

F50.

78 Loop 0.

78 Loop 0.

72 Loop 0.

X-.1 Y1.5.

F15, Z 1st 0, Z inc. 1, R.1, Z depth 0.

78 Loop 0.

83 Loop 0.

72 Loop 0.

98 Loop 0.

EIA - Mode Program 1

1001 G300 P.1

R-Plane

1002 M600 F50 X-1.6 Y1.5

Start Point

1003 G878 G878 G872

Letters IIC

1004 X-.1 Y1.5

Space

1005 G878 G883 G872 G898

Letters INC.

Help - Program 2

Screens

Initialize

Comp. Subs.

Move

Rotate

Move

Comp. Subs.

Cut-Subroutine

Rotate

Move

Comp. Sub

Cut-Subroutine

Rotate

Move

Comp. Sub.

Cut-Subroutine

Rotate

Move

Comp. Sub.

Cut-Subroutine

Data

ABS, Inch X=.5, Y=.5, Z=1, A=0

All

X=0, Y=0

A=110 degrees

X0 Y2

61

61 Loop 0

100

X0 Y2

64

64 Loop 0

80

X0 Y2

66

66 Loop 0

63

X0 Y2

62

62 Loop 0

Rotate 110 degrees

Twice Radius

Applies scaling, and
rotation to sub 61

Call sub. 61 the #1

#4

#6

Rotate
Move
Comp. Sub.
Subroutine

Repeat this sequence for all
desired letters.

EIA - Program 2

2001 G300 P.1
G393 P0
G384 P1
G385 P1
X0 Y0
G384 P.5
G385 P.5
G393 A110
X0 Y2
M661 G861

R-Plane
cancel scaling
and
rotation
Center of arc.
 $\frac{1}{2}$ scale X
 $\frac{1}{4}$ scale Y
Rotate 110 degrees.
Radius of letters.
M661 applies scaling and rotation to
sub. 61, G861 calls sub. 61 the num-
ber 1.

G393 A100
X0 Y2
M664 G864

#4

G393 A80
X0 Y2
M666 G866

#6

G393 A63
X0 Y2
M662 G862

#2

G393 A45
X0 Y2
M667 G867

#7

G393 A25
X0 Y2
M682 G882

M

G893 A00
X0 Y2
M670 G870

A

G393 A-20
X0 Y2
M687 G887

R

G393 A-35
X0 Y2
M689 G889

T

G393 A-48
X0 Y2
M678 G878

I

G393	A-55	
X0	Y2	N
M683	G883	
G393	A-80	
X0	Y2	D
M673	G873	
G393	A-95	
X0	Y2	R
M687	G887	
G393	A-115	
X0	Y2	.
M698	G898	

Note that the angular displacement of the letters varies as the letters progress around the circle. This is due to the varying width of the letters and their angle of rotation. The minimum rotation angle should be the 12 o'clock position and the letters should be rotated + and - from this point.

To use this tape simply place the tape into the reader and load the alphabet section. Construct a main program similar to programs 1 and 2 calling the desired letter subroutines. The letters can be scaled up or down and rotated by changing parameters 84, 85, and 93 in the main program. The following is a list of the subroutines and their letters.

SUBROUTINE TABLE

<u>Subroutine Number</u>	<u>Event Number</u>	<u>Letter or Symbol</u>
70	1	A
71	2	B
72	3	C
73	4	D
74	5	E
75	6	F
76	7	G
77	8	H
78	9	I
79	10	J
80	11	K
81	12	L
82	13	M
83	14	N
84	15	O
85	16	P
86	17	Q
87	18	R
88	19	S
89	20	T
90	21	U
91	22	V
92	23	W
93	24	X
94	25	Y
95	26	Z
96	37	-
97	38	+
98	39	.
99	40	,
60	36	0
61	27	1
62	28	2
63	29	3
64	30	4
65	31	5
66	32	6
67	33	7
68	34	8
69	35	9

NOTE: If the tape fills up too much memory, the unused letters can be deleted. To do this use the display-edit program features of the help mode. Select program 9, skip tool #, event to be deleted, skip operation #, and select Delete Event.

