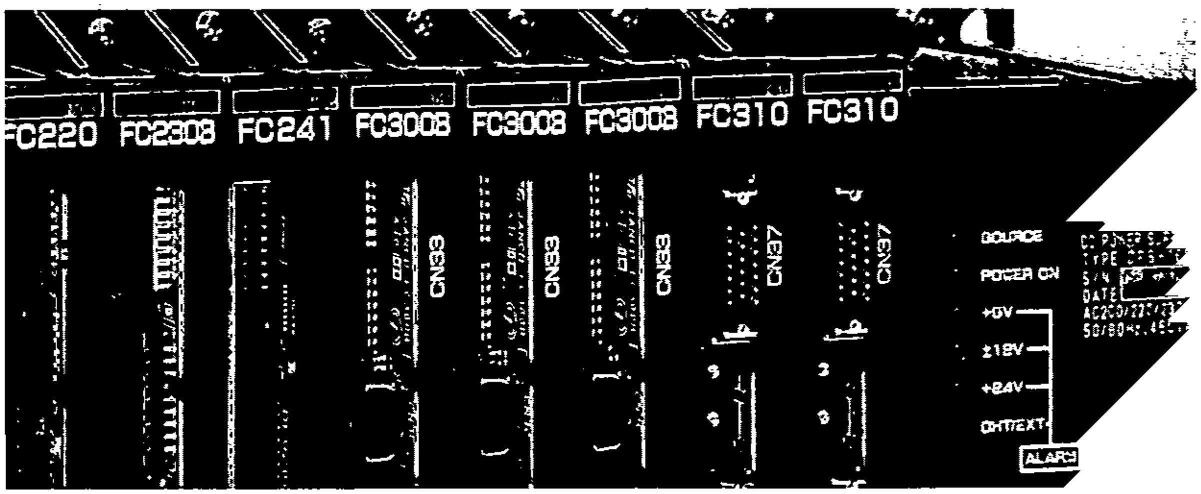
YASKAWA

YASNAC i80L

INSTRUCTIONS

CNC SYSTEM FOR TURNING APPLICATIONS



Before initial operation, read these instructions thoroughly, and retain for future reference.

YASKAWA

TOE-C843-11.20D

PREFACE

This manual contains information such as notes, basic configuration, programming, functions, and operation required to use **YASNAC i80L** [NC operation panel with 9″ CRT (basic)].

Easy-to-read items summarized in a list format are found in the appendix supplied with this manual.

This manual and the appendix (TOE-C843-11.21) should be read in conjunction with each other.

This manual will help you to take full advantage of the functions of **YASNAC i80L**.

NOTES FOR YASNAC i80L OPERATOR'S MANUAL:

1. Optional features are marked with ∗. For the specifications of your **YASNAC i80L**, refer to the machine tool builder's manuals.
2. Unless otherwise specified, the following rules apply to the descriptions of programming examples shown in this manual.
   - Dimensions: in mm
   - Absolute Zero Point: ⯁
   - Reference Zero Point: ⯁
3. Operations not described in this manual should not be attempted with the control. If you have any questions, contact your YASKAWA representative.
4. The functions and performance as an NC machine are determined by a combination of a machine and the NC control. For operation of your NC machine, the machine tool builder's manual shall take priority over this manual.

CONTENTS

1 OVERVIEW / 1

2 PROGRAMMING / 5

3 DESCRIPTION OF OPERATION / 347

CONTENTS

Page

1 OVERVIEW 1
1.1 BASIC CONFIGURATION OF NC MACHINE TOOL SYSTEM 2
1.2 BASIC OPERATION 3
1.3 MULTI-PROGRAM CONTROL (only when the multi-program option is provided) 4

2 PROGRAMMING 5
2.1 CONTROL AXES 7
2.1.1 Control Axis and Number of Simultaneously Controllable Axes 7
2.1.2 Input Increment and Output Increment 8
2.1.2.1 Least input increment and least input increment × 10 8
2.1.2.2 Least output increment 8
2.1.3 Maximum Move Command Values 9
2.2 PROGRAMMING 10
2.2.1 Process Sheet 10
2.2.2 Programming Format 10
2.2.2.1 Label part and label skip 12
2.2.2.2 Tape start/tape end 12
2.2.2.3 Progarm start/program end 13
2.2.2.4 Progarm part 14
2.2.2.5 Comment part 22
2.2.3 Buffer Register and Multi-active Register 24
2.2.4 Tape Code 25
2.2.4.1 Tape codes 25
2.2.4.2 EIA/ISO auto-select 25
2.2.5 NC Tape 27
2.2.5.1 Paper tape 27
2.2.5.2 NC tape punch 27
2.2.5.3 NC tape check 27
2.2.5.4 Splicing NC tape 28
2.2.5.5 Keeping NC tape 28
2.3 PREPARATORY FUNCTION (G-FUNCTION) 29
2.4 INTERPOLATION 33
2.4.1 Positioning 33
2.4.1.1 G00 (modal G code of 01 group) 33
2.4.1.2 G06 (non-modal G code of ✻ group) 34
2.4.2 Linear Interpolation (G01) 35
2.4.3 Chamfering Corner Radius Designation (G011, G12)✻ 39
2.4:3.1 Chamfering command (G11) 39
2.4.3.2 Rounding designation (G12) 40
2.4.4 Circular Interpolation (G02, G03, G22, G23) 42
2.4.5 Polar Coordinate Interpolation (G126, G127)✻ 46
2.4.6 Cylindrical Interpolation (G124, G125)✻ 51
2.4.7 Tool Diameter Compensation Functions Available in Polar Coordinate Interpolation and Cylindrical Interpolation 54

Page

2.5 FEED FUNCTIONS 56
2.5.1 Rapid Traverse Rate 56
2.5.2 Cutting Feed 56
2.5.2.1 Feed per rotation (G99 mode) 57
2.5.2.2 Feed per minute (G98 Mode) 59

2.5.3 Switching between Feed per Minute and Feed per Rotation (G98/G99) 62
2.5.4 Screw Cutting and Continuous Screw Cutting (G32) 63
2.5.5 Multiple Thread Screw Cutting (G32)✻ 68
2.5.6 Variable Lead Screw Cutting (G34)✻ 71
2.5.7 Automatic Acceleration and Deceleration 74

2.5.7.1 Accel/decel of rapid traverse and manual feed 74
2.5.7.2 Accel/decel of cutting feed 75
2.5.8 Override 76
2.5.8.1 Cutting feed override 76
2.5.8.2 Rapid traverse override 77

2.5.9 Dwell (G04) 78
2.5.9.1 Dwell per minute 78
2.5.10 Speed Control Command 79

2.6 REFERENCE POINT RETURN 80
2.6.1 Automatic Reference Point Return (G28) 80
2.6.2 Reference Point Return Check 85
2.6.3 Return from Reference Point Return (G29) 86
2.6.4 Second to Fourth Reference Point Return (G30)✻ 89

2.7 COORDINATE SYSTEM 90
2.7.1 Setting Coordinate System 90
2.7.2 Automatic Setting of Coordinate System 93
2.7.3 Shifting Work Coordinate System 94
2.7.4 Setting Work Coordinate System (G50T, G51)✻ 96

2.8 ENTERING COORDINATE VALUES 103
2.8.1 Absolute and Incremental Commands 103
2.8.2 Selection of Diameter Specification and Radius Specification of X-axis 105
2.8.3 Inch/Metric Input Specification (G20, G21) 106
2.8.4 Decimal Point Input 108

2.9 SPINDLE FUNCTION (S 5-DIGIT SPECIFICATION) 109
2.9.1 Spindle Command 109
2.9.2 Spindle Maximum Rotation Speed Command (G50S) 111
2.9.3 Peripheral Speed Keeping Control (G93, G97)✻ 112
2.9.3.1 Peripheral speed keeping control (G96)✻ 112

2.9.3.2 Cancelling peripheral speed keeping control (G97)✻ 113
2.9.4 Rotary Tool Spindle Selection Function ✻ 116

2.10 TOOL FUNCTION (T FUNCTION) 117
2.10.1 T 4-digit Specification 117

CONTENTS (Cont'd)

Page

2.10.2 T 6-digit Specification✽ 117

2.11 MISCELLANEOUS FUNCTION (M FUNCTION) 118
2.11.1 M Function 118
2.11.1.1 M codes related to stopping (M00, M01, M02, M30) 118
2.11.1.2 Internal processing M codes 119
2.11.1.3 General other M codes 120

2.11.1.4 Move than one M command in a block ✽ 120

2.12 TOOL OFFSET 121
2.12.1 Offset Memory 121
2.12.2 Tool Position Offset 123
2.12.3 Nose R Compensation (G40, G41/G42)✽ 128
2.12.4 Multi-active Register (M92, M93)✽ 172

2.12.5 Corner Rounding Judgement Internal M Code (M96/M97) 173

2.13 PROGRAM SUPPORT FUNCTIONS 174
2.13.1 Canned Cycle (G90, G92, G94) 174
2.13.1.1 Cutting cycle A (G90) 175
2.13.1.2 Thread cutting cycle (G92) 178
2.13.1.3 Thread cutting cycle (G94) 185

2.13.2 Multiple Repetitive Cycle (G70 to G76)✽ 188
2.13.2.1 Overview 188
2.13.2.2 Outer diameter rough cutting cycle (G71) 190
2.13.2.3 End face rough cutting cycle (G72) 200
2.13.2.4 Closed loop cutting cycle (G73) 205

2.13.2.5 Finishing cycle (G70) 209
2.13.2.6 End face cutting off cycle (G74) 212
2.13.2.7 Outer diameter cutting cycle (G75) 215
2.13.2.8 Automatic thread cutting cycle (G76) 218
2.13.2.9 Remarks on G70 to G78 224

2.13.3 Complex Chamfering (G111, G112)✽ 227
2.13.3.1 Taper complex chamfering, rounding command (G111)✽ 227
2.13.3.2 Circular complex chamfering, rounding command (G112)✽ 240
2.13.4 Tool Offset Amount Setting (G10)✽ 248
2.13.5 Progarm Call Function 250

2.13.5.1 Subprogarm (M98, M99) 250
2.13.6 Drilling Canned Cycle ✽ 252
2.13.6.1 Drilling canned cycle✽ 252

2.14 MACRO PROGRAM (G65, G66, G67)✽ 267
2.14.1 Macro Progarm Call 269
2.14.1.1 Macro nesting call 274
2.14.1.2 Argument designation 276
2.14.2 Variables 280

2.14.3 Spacifying Variables 297
2.14.4 Quoting Variables 298

Page

2.14.5 Undefined Variables 299
2.14.6 Arithmetic Commands 300
2.14.7 Control Commands 303
2.14.8 Entering Macro Programs 308
2.14.9 RS-232C Data Output-2 (BPRNT, DPRNT) 309
2.14.10 Macro Program Alarm Numbers 314
2.14.11 Macro Program Examples 315
2.15 AUTO MEASUREMENT FUNCTION 322
2.15.1 Skip Function (G31) * 322
2.16 STORED STROKE LIMIT CHECK 324
2.16.1 Stored Stroke Limit A 324
2.16.2 Stored Stroke Limit B 325
2.17 MULTI-SELECTION SUPPORT FUNCTION (FOR MULTI-SECTION OPTION ONLY) 332
2.17.1 Section Synchronization Command * 332
2.17.2 Password Command 337
2.17.3 Control Axis Select Command 340
2.17.4 Local Section Spindle Selection Function * 341
2.17.5 Synchronous Feed Reference Spindle Select * 343
2.17.6 Remote Program M, T Command * 344
2.17.7 Pulse Copy Function * 345

3 DESCRIPTION OF OPERATION 347
3.1 OUTLINE OF SCREENS 350
3.2 CONFIGURATION OF PROCESSES, JOBS, AND FUNCTIONS 351
3.3 NC OPERATOR FOR PANEL AND DISPLAY SCREEN 354
3.3.1 Outline of Operator Panel Key Functions 354
3.3.1.1 Power on/off buttons 354
3.3.1.2 9-inch graphic display 355
3.3.1.3 Process keys 356
3.3.1.4 Soft keys 356
3.3.1.5 Address keys 358
3.3.1.6 Data keys 359
3.3.1.7 Page keys 359
3.3.1.8 Cursor keys 360
3.3.1.9 Action keys 361
3.3.1.10 Reset key 362
3.3.1.11 AUX keys 362
3.4 DISPLAY AND WRITE 363
3.4.1 Constant Display 363
3.4.2 Pop-up Menu 368
3.4.3 Key Buffer Editing Function 369
3.4.4 Buzzer Function 370
3.4.5 $\boxed{\text{PROG}}$ (Program Editing) Process 371

CONTENTS (Cont'd)

Page

3.4.5.1 Part program editing 372
3.4.5.2 Part program directory job (job soft key [DIR]) 395
3.4.5.3 Part program I/O verification ([IN/OUT] job soft key) 411

3.4.6 SET Setup Process 439

3.4.6.1 Offset function 440
3.4.6.2 Work coordinate system shift function 445

3.4.7 Run Process RUN 447

3.4.7.1 Program job 448
3.4.7.2 Command value job (COMMAND VALUE job soft key) 463
3.4.7.3 Setting job 464

3.4.8 Maintenance Process MAINT 471

3.4.8.1 Parameter job 472
3.4.8.2 I/O monitor job 479
3.4.8.3 I/O verification job 485

3.4.9 Common Process COMN 491

3.4.9.1 Absolute position job (POS. job soft key) 492
3.4.9.2 Alarm job (ALARM job soft key) 500
3.4.9.3 Time job (TIME job soft key) 504
3.4.9.10 Power-on Initial Indication 509

INDEX

	Subject	Par.	Page
	9-inch graphic display	3.3.1.2	355
A	Absolute and Incremental Commands	2.8.1	103
	Absolute position job (POS. job soft key)	3.4.9.1	493
	Accel/decel of rapid traverse and manual feed.	2.5.7.1	74
	Accel/decel of cutting feed	2.5.7.2	75
	Action keys	3.3.1.9	361
	Address keys	3.3.1.5	358
	Alarm job (ALARM job soft key)	3.4.9.2	500
	Argument designation	2.14.1.2	276
	Arithmetic Commands	2.14.6	300
	AUTO MEASUREMENT FUNCTION	2.15	322
	Automatic Acceleration and Deceleration	2.5.7	74
	Automatic Reference Point Return (G28)	2.6.1	80
	Automatic Setting of Coordinate System	2.7.2	93
	Automatic thread cutting cycle (G76)	2.13.2.8	218
	AUX keys	3.3.1.11	362
B	BASIC CONFIGURATION OF NC MACHINE TOOL SYSTEM	1.1	2
	BASIC OPERATION	1.2	3
	Buffer Register and Multi-active Register	2.2.3	24
	Buzzer Function	3.4.4	370
C	Cancelling peripheral speed keeping control (G97)✻	2.9.3.2	113
	Canned Cycle (G90, G92, G94)	2.13.1	174
	Chamfering Command (G11)	2.4.3.1	39
	Chamfering corner Radius Designation (G011, G12)✻	2.4.3	39
	Circular complex chamfering, rounding command (G112)✻	2.13.3.2	240
	Circular Interpolation (G02, G03, G22, G23)	2.4.4	42
	Closed loop cutting cycle (G73)	2.13.2.4	205
	Command value job (COMMAND VALUE job soft key)	3.4.7.2	463
	Comment part	2.2.2.5	22
	Common Process (COMN)	3.4.9	491
	Complex Chamfering (G111, G112)✻	2.13.3	227
	CONFIGURATION OF PROCESSES, JOBS, AND FUNCTIONS	3.2	351
	Constant Display	3.4.1	363
	CONTROL AXES	2.1	7
	Control Axis and Number of Simultaneously Controllable Axes	2.1.1	7
	Control Axis Select Command	2.17.3	340
	Control Commands	2.14.7	303
	COORDINATE SYSTEM	2.7	90
	Corner Rounding Judgement Internal M Code (M96/M97)	2.12.5	173
	Cursor keys	3.3.1.8	360
	Cutting cycle A (G90)	2.13.1.1.	175
	Cutting feed override	2.5.8.1	76
	Cutting Feed	2.5.2	56
	Cylindrical Interpolation (G124, G125)✻	2.4.6	51
D	Data keys	3.3.1.6	359
	Decimal Point Input	2.8.4	108
	DESCRIPTION OF OPERATION	3	347
	DISPLAY AND WRITE	3.4	363
	Drilling Canned Cycle ✻	2.13.6	252

INDEX (Cont'd)

	Subject	Par.	Page
D	Drilling canned cycle ✻	2.13.6.1	252
	Dwell (G04)	2.5.9	78
	Dwell per minute	2.5.9.1	78
E	EIA/ISO auto-select	2.2.4.2	25
	End face cutting off cycle (G74)	2.13.2.6	212
	End face rough cutting cycle (G72)	2.13.2.3	200
	Entering Macro Programs	2.14.8	308
	ENTERING COORDINATE VALUES	2.8	103
F	FEED FUNCTIONS	2.5	56
	Feed per minute (G98 Mode)	2.5.2.2	59
	Feed per rotation (G99 mode)	2.5.2.1	57
	Finishing cycle (G70)	2.13.2.5	209
G	G00 (modal G code of 01 group)	2.4.1.1	33
	G06 (non-modal G code of ✻ group)	2.4.1.2	34
	General other M codes	2.11.1.3	120
I	I/O monitor job	3.4.8.2	479
	I/O verification job	3.4.8.3	485
	Inch/Metric Input Specification (G20, G21)	2.8.3	106
	Input Increment and Output Increment	2.1.2	8
	Internal processing M codes	2.11.1.2	119
	INTERPOLATION	2.4	33
K	Keeping NC tape	2.2.5.5	28
	Key Buffer Editing Function	3.4.3	369
L	Label part and label skip	2.2.2.1	12
	Least input increment and least input increment ×10	2.1.2.1	8
	Least output increment	2.1.2.2	8
	Linear Interpolation (G01)	2.4.2	35
	Local Section Spindle Selection Function ✻	2.17.4	341
M	M codes related to stopping (M00, M01, M02, M30)	2.11.1.1	118
	M Function	2.11.1	118
	Macro nesting call	2.14.1.1	274
	Macro Progarm Call	2.14.1	269
	MACRO PROGRAM (G65, G66, G67)✻	2.14	267
	Macro Program Alarm Numbers	2.14.10	314
	Macro Program Examples	2.14.11	315
	Maintenance Process MAINT	3.4.8	471
	Maximum Move Command Values	2.1.3	9
	MISCELLANEOUS FUNCTION (M FUNCTION)	2.11	118
	Move than one M command in a block ✻	2.11.1.4	120
	MULTI-PROGRAM CONTROL (only when the multi-program option is provided)	1.3	4
	MULTI-SELECTION SUPPORT FUNCTION (FOR MULTI-SECTION OPTION ONLY)	2.17	332
	Multiple Repetitive Cycle (G70 to G76)✻	2.13.2	188
	Multiple Thread Screw Cutting (G32)✻	2.5.5	68
	Multi-active Register (M92, M93)✻	2.12.4	172
N	NC OPERATOR FOR PANEL AND DISPLAY SCREEN	3.3	354
	NC tape check	2.2.5.3	27
	NC Tape	2.2.5	27
	NC tape punch	2.2.5.2	2
	Nose R Compensation (G40, G41/G42)✻	2.12.3	128

	Subject	Par.	Page
O	Offset function	3.4.6.1	440
	Offset Memory	2.12.1	121
	Outer diameter cutting cycle (G75)	2.13.2.7	215
	Outer diameter rough cutting cycle (G71)	2.13.2.2	190
	Outline of Operator Panel Key Functions	3.3.1	354
	OUTLINE OF SCREENS	3.1	350
	Override	2.5.8	76
	OVERVIEW	1	1
	Overview	2.13.2.1	188
P	Page keys	3.3.1.7	359
	Paper tape	2.2.5.1	27
	Parameter job	3.4.8.1	472
	Part program directory job (job soft key DIR)	3.4.5.2	395
	Part program editing	3.4.5.1	372
	Part program I/O verification (IN/OUT job soft key)	3.4.5.3	411
	Password Command	2.17.2	337
	Peripheral Speed Keeping Control (G93, G97)*	2.9.3	112
	Peripheral speed keeping control (G96) *	2.9.3.1	112
	Polar Coordinate Interpolation (G126, G127)*	2.4.5	46
	Positioning	2.4.1	33
	Power-on Initial Indication	3.4.10	509
	PREPARATORY FUNCTION (G-FUNCTION)	2.3	29
	Process keys	3.3.1.3	356
	Process Sheet	2.2.1	10
	Progarm Call Function	2.13.5	250
	Progarm part	2.2.2.4	14
	PROC (Program Editing) Process	3.4.5	371
	Progarm start/program end	2.2.2.3	13
	PROGRAM SUPPORT FUNCTIONS	2.13	174
	Program job	3.4.7.1	448
	PROGRAMMING	2	5
	Programming Format	2.2.2	10
	Pulse Copy Function*	2.17.7	345
	Pop-up Menu	3.4.2	368
Q	Quoting Variables	2.14.4	298
R	Rapid traverse override	2.5.8.2	77
	Rapid Traverse Rate	2.5.1	56
	REFERENCE POINT RETURN	2.6	80
	Reference Point Return Check	2.6.2	85
	Remarks on G70 to G78	2.13.2.9	224
	Remote Program M, T Command *	2.17.6	344
	Reset key	3.3.1.10	362
	Return from Reference Point Return (G29)	2.6.3	86
	Rotary Tool Spindle Selection Function *	2.9.4	116
	Rounding designation (G12)	2.4.3.2	40
	RS-232C Data Output-2 (BPRNT, DPRNT)	2.14.9	310
	Run Process RUN	3.4.7	447

INDEX (Cont'd)

	Subject	Par.	Page
S	Screw Cutting and Continuous Screw Cutting (G32)*	2.5.4	63
	Second to Fourth Reference Point Return (G30)	2.6.4	89
	Section Synchronization Command *	2.7.1	332
	Selection of Diameter Specification and Radius Specification of X-axis	2.8.2	105
	Setting Coordinate System	2.7.1	90
	Setting job	3.4.7.3	464
	Setting Work Coordinate System (G50T, G51)*	2.7.4	96
	[SET] Setup Process	3.4.6	439
	Shifting Work Coordinate System	2.7.3	94
	Skip Function (G31)*	2.15.1	322
	Soft keys	3.3.1.4	356
	Specifying Variables	2.14.3	297
	Speed Control Command	2.5.10	79
	Spindle Command	2.9.1	109
	SPINDLE FUNCTION (S 5-DIGIT SPECIFICATION)	2.9	109
	Spindle Maximum Rotation Speed Command (G50S)	2.9.2	111
	Splicing NC tape	2.2.5.4	28
	Stored Stroke Limit A	2.16.1	324
	Stored Stroke Limit B	2.16.2	324
	STORED STROKE LIMIT CHECK	2.16	325
	Subprogarm (M98, M99)	2.13.5.1	250
	Switching between Feed per Minute and Feed per Rotation (G98/G99)	2.5.3	62
	Synchronous Feed Reference Spindle Select *	2.17.5	343
T	T 4-digit Specification	2.10.1	117
	T 6-digit Specification	2.10.2	117
	Tape Code	2.2.4	25
	Tape codes	2.2.4.1	25
	Tape start/tape end	2.2.2.2	12
	Taper complex chamfering« rounding command (G111)*	2.13.3.1	227
	Thread cutting cycle (G92)	2.13.1.2	178
	Thread cutting cycle (G94)	2.13.1.3	185
	Time job ([TIME] job soft key)	3.4.9.3	504
	Tool Diameter Compensation Functions Available in Polar Coordinate Interpolation and Cylindrical Interpolation	2.4.7	54
	TOOL FUNCTION (T FUNCTION)*	2.10	117
	Tool Offset Amount Setting (G10)*	2.13.4	248
	TOOL OFFSET	2.12	121
	Tool Position Offset	2.12.2	123
U	Undefined Variables	2.14.5	299
V	Variable Lead Screw Cutting (G34)*	2.5.6	71
	Variables	2.14.2	280
W	Work coordinate system shift function	3.4.6.2	445

SECTION 1 OVERVIEW

This section describes the basic configuration and operation of the NC machine tool system.

SECTION 1
CONTENTS

Page

1 OVERVIEW 1

1.1 BASIC CONFIGURATION OF NC MACHINE TOOL SYSTEM 2

1.2 BASIC OPERATION 3

1.3 MULTI-PROGRAM CONTROL (only when the multi-program option is provided) 4

1.1 BASIC CONFIGURATION OF NC MACHINE TOOL SYSTEM

YASNAC i80L allows parallel processing of multiple CPUs organically by applying digital servo drives and actuators, resulting in high speed and high precision processing.

Fig. 1.1 shows the basic configuration of the NC machine tool system. Each control part and control processing part is incorporated in one or several CPUs. These parts execute parallel processing except for requirements for synchronization with specified software and/or hardware.

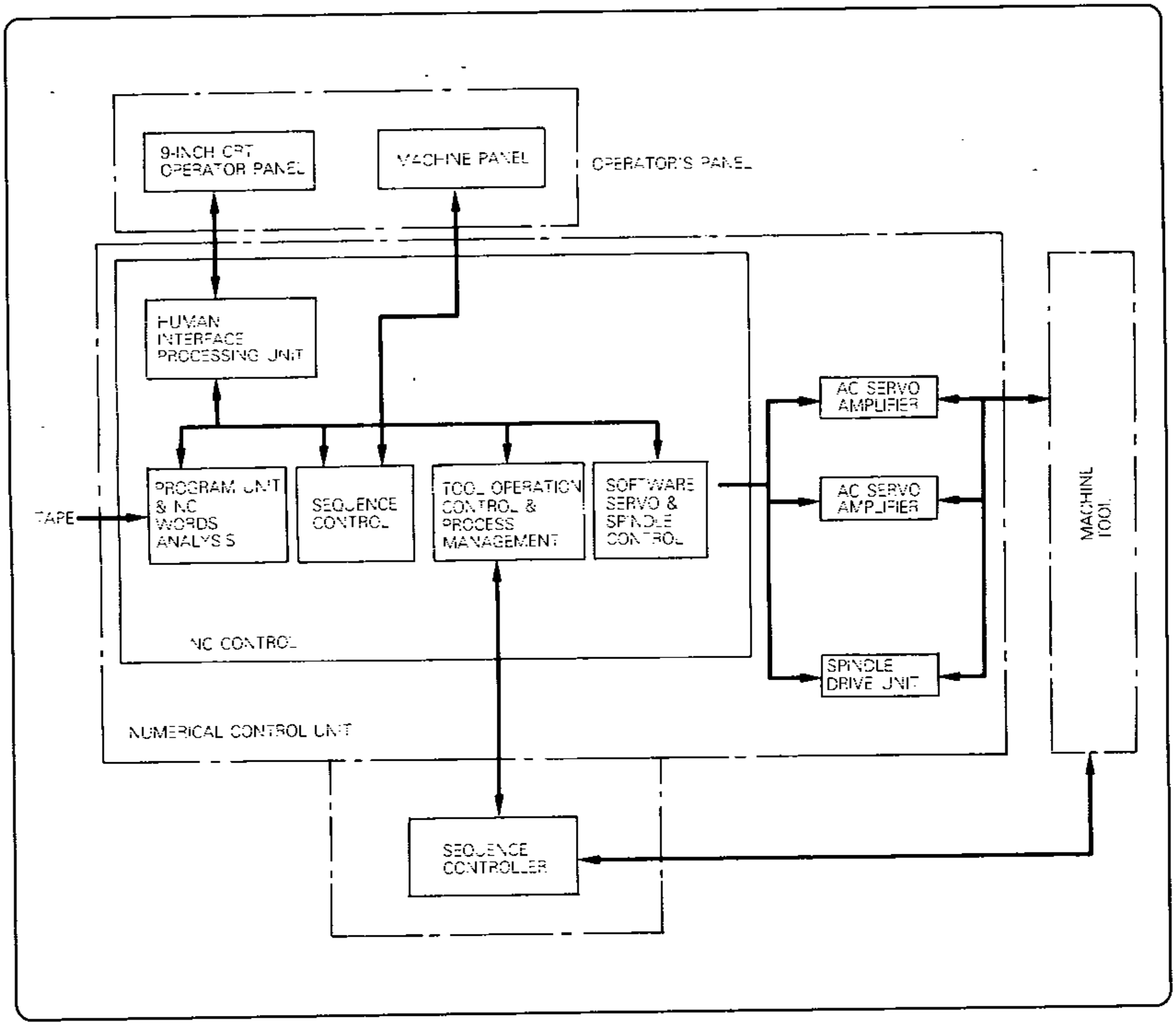


Fig. 1.1 Basic Configuration of NC Machine Tool System with YASNAC i80L

1.2 BASIC OPERATION

To process parts with NC machine tools, it will first be necessary to prepare a processing program, input this in the NC, set the work coordinates required to execute this program (run the machine), make the necessary preparations such as setting tool offset and editing, and then depress the START button on the machine operation panel to proceed with the processing work.

A general flow of this system is shown in **Fig. 1.2**.

A general configuration of this manual is shown on the right side of this flow. **Section 2, "PROGRAMMING"** explains the functions of the control unit, the instruction words usable in the program, and their use.

Section 3, "OPERATION" explains in detail how to use the machine tool system and the method of operating the operator panel with 9″ CRT and the machine operation panel (reference figure).

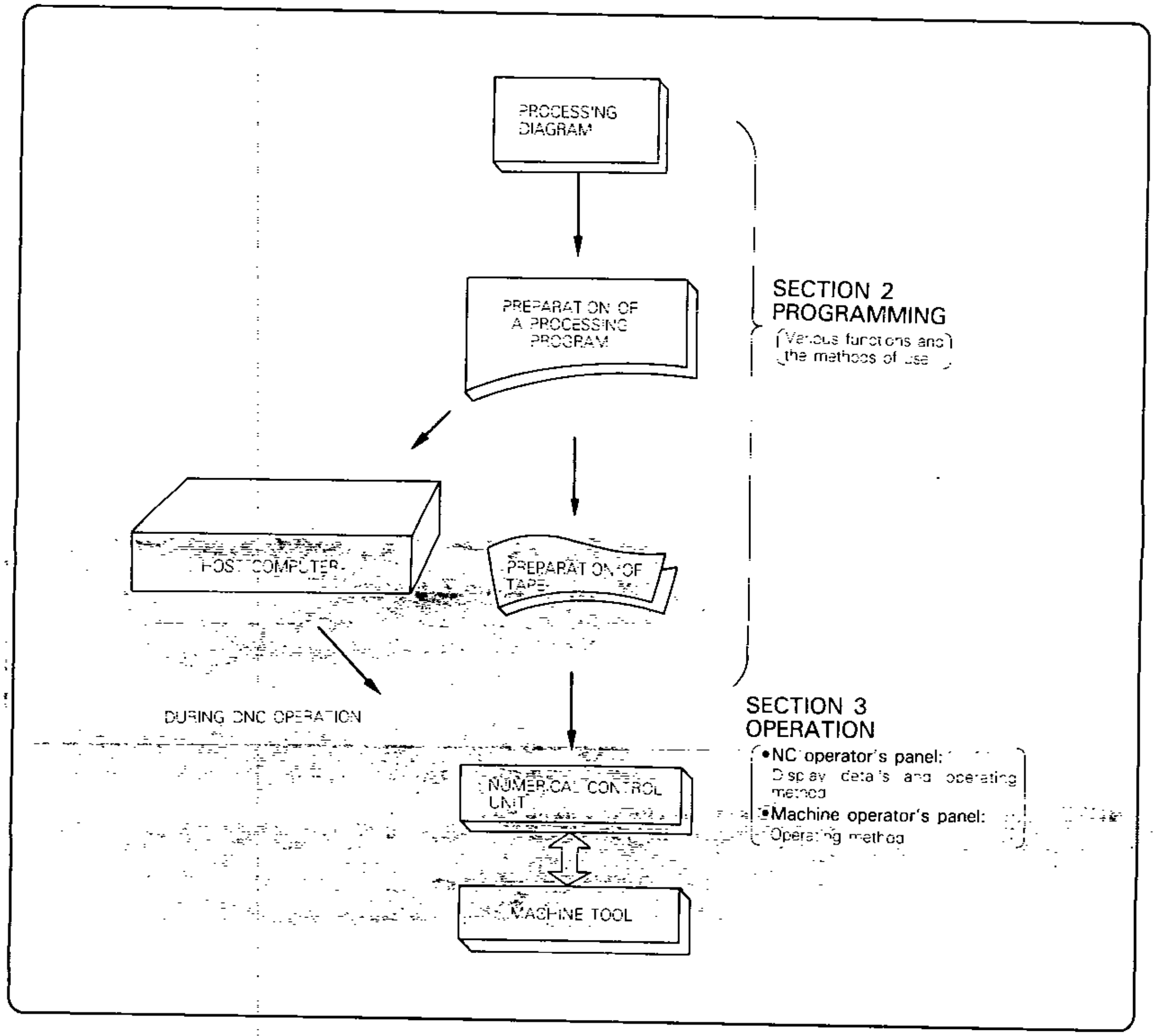


Fig. 1.2

1.3 MULTI-PROGRAM CONTROL (only when the multi-program option is provided)

The environment in which NC processing programs are executed and servo axes are controlled is called a program.

YASNAC i80L allows up to three programs to operate at the same time under compound turning machines.

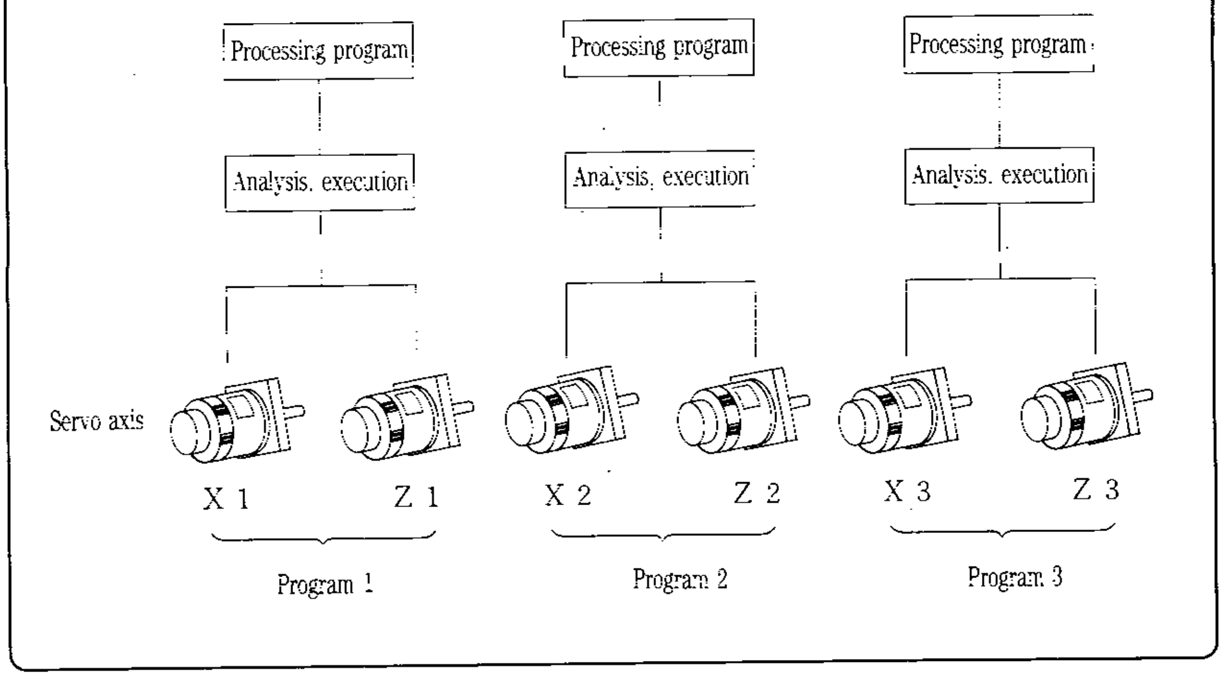
Individual programs can execute one processing program independently.

The program can also perform processing in conjunction with other programs, thereby enabling the processing that could not be performed in a single program.

The number of available programs and varied operations depend on machines. For details, refer to command or reference manuals issued by your machine tool manufacturer.

Section 2 assumes a single program for explanation purposes unless otherwise noted.

Expmple of Programming



NOTE The multi-program control function is provided as an option.

SECTION 2 PROGRAMMING

This section describes programming and functions [(including optional functions (∗1)] of the control axes.

∗1 Optional functions are marked with ∗.

SECTION 2 CONTENTS

	Page
2 PROGRAMMING	5
2.1 CONTROL AXES	7
2.1.1 Control Axis and Number of Simultaneously Controllable Axes	7
2.1.2 Input Increment and Output Increment	8
2.1.2.1 Least input increment and least input increment × 10	8
2.1.2.2 Least output increment	8
2.1.3 Maximum Move Command Values	9
2.2 PROGRAMMING	10
2.2.1 Process Sheet	10
2.2.2 Programming Format	10
2.2.2.1 Label part and label skip	12
2.2.2.2 Tape start/tape end	12
2.2.2.3 Progarm start/program end	13
2.2.2.4 Progarm part	14
2.2.2.5 Comment part	22
2.2.3 Buffer Register and Multi-active Register	24
2.2.4 Tape Code	25
2.2.4.1 Tape codes	25
2.2.4.2 EIA/ISO auto-select	25
2.2.5 NC Tape	27
2.2.5.1 Paper tape	27
2.2.5.2 NC tape punch	27
2.2.5.3 NC tape check	27
2.2.5.4 Splicing NC tape	28
2.2.5.5 Keeping NC tape	28
2.3 PREPARATORY FUNCTION (G-FUNCTION)	29
2.4 INTERPOLATION	33
2.4.1 Position	33
2.4.1.1 G00 (modal G code of 01 group)	33
2.4.1.2 G06 (non-modal G code of ∗ group)	34
2.4.2 Linear Interpolation (G01)	35
2.4.3 Chamfering Corner Radius Designation (G011, G12)∗	39
2.4.3.1 Chamfering command (G11)	39
2.4.3.2 Rounding designation (G12)	40
2.4.4 Circular Interpolation (G02, G03, G22, G23)	42
2.4.5 Polar Coordinate Interpolation (G126, G127)∗	46
2.4.6 Cylindrical Interpolation (G124, G125)∗	51
2.4.7 Tool Diameter Compensation Functions Available in Polar Coordinate Interpolation and Cylindrical Interpolation	54
2.5 FEED FUNCTIONS	56
2.5.1 Rapid Traverse Rate	56
2.5.2 Cutting Feed	56
2.5.2.1 Feed per rotation (G99 mode)	57
2.5.2.2 Feed per minute (G98 Mode)	59
2.5.3 Switching between Feed per Minute and Feed per Rotation (G98/G99)	62
2.5.4 Screw Cutting and Continuous Screw Cutting (G32)	63
2.5.5 Multiple Thread Screw Cutting (G32)∗	68
2.5.6 Variable Lead Screw Cutting (G34)	71
2.5.7 Automatic Acceleration and Deceleration	74
2.5.7.1 Accel/decel of rapid traverse and manual feed	74
2.5.7.2 Accel/decel of cutting feed	75
2.5.8 Override	76
2.5.8.1 Cutting feed override	76
2.5.8.2 Rapid traverse override	77
2.5.9 Dwell (G04)	78
2.5.9.1 Dwell per minute	78
2.5.10 Speed Control Command	79
2.6 REFERENCE POINT RETURN	80
2.6.1 Automatic Reference Point Return (G28)	80
2.6.2 Reference Point Return Check	85
2.6.3 Return from Reference Point Return (G29)	86
2.6.4 Second to Fourth Reference Point Return (G30)∗	89
2.7 COORDINATE SYSTEM	90
2.7.1 Setting Coordinate System	90
2.7.2 Automatic Setting of Coordinate System	93
2.7.3 Shifting Work Coordinate System	94
2.7.4 Setting Work Coordinate System (G50T, G51)∗	96
2.8 ENTERING COORDINATE VALUES	103
2.8.1 Absolute and Incremental Commands	103
2.8.2 Selection of Diameter Specification and Radius Specification of X-axis	105
2.8.3 Inch/Metric Input Specification (G20, G21)	106
2.8.4 Decimal Point Input	108

SECTION 2
CONTENTS (Cont'd)

	Page
2.9 SPINDLE FUNCTION (S 5-DIGIT SPECIFICATION)	109
2.9.1 Spindle Command	109
2.9.2 Spindle Maximum Rotation Speed Command (G50S)	111
2.9.3 Peripheral Speed Keeping Control (G93, G97)*	112
2.9.3.1 Peripheral speed keeping control (G96)*	112
2.9.3.2 Cancelling peripheral speed keeping control (G97)*	113
2.9.4 Rotary Tool Spindle Selection Function *	116
2.10 TOOL FUNCTION (T FUNCTION)	117
2.10.1 T 4-digit Specification	117
2.10.2 T 6-digit Specification*	117
2.11 MISCELLANEOUS FUNCTION (M FUNCTION)	118
2.11.1 M Function	118
2.11.1.1 M codes related to stopping (M00, M01, M02, M30)	118
2.11.1.2 Internal processing M codes	119
2.11.1.3 General other M codes	120
2.11.1.4 Move than one M command in a block	120
2.12 TOOL OFFSET	121
2.12.1 Offset Memory	121
2.12.2 Tool Position Offset	123
2.12.3 Nose R Compensation (G40, G41/G42)*	128
2.12.4 Multi-active Register (M92, M93)*	172
2.12.5 Corner Rounding Judgement Internal M Code (M96/M97)	173
2.13 PROGRAM SUPPORT FUNCTIONS	174
2.13.1 Canned Cycle (G90, G92, G94)	174
2.13.1.1 Cutting cycle A (G90)	175
2.13.1.2 Thread cutting cycle (G92)	178
2.13.1.3 Thread cutting cycle (G94)	185
2.13.2 Multiple Repetitive Cycle (G70 to G76)*	188
2.13.2.1 Overview	188
2.13.2.2 Outer diameter rough cutting cycle (G71)	190
2.13.2.3 End face rough cutting cycle (G72)	200
2.13.2.4 Closed loop cutting cycle (G73)	205
2.13.2.5 Finishing cycle (G70)	209
2.13.2.6 End face cutting off cycle (G74)	212
2.13.2.7 Outer diameter cutting cycle (G75)	215
2.13.2.8 Automatic thread cutting cycle (G76)	218
2.13.2.9 Remarks on G70 to G78	224
2.13.3 Complex Chamfering (G111, G112) *	227
2.13.3.1 Taper complex chamfering, rounding command (G111)*	227
2.13.3.2 Circular complex chamfering, rounding command (G112)*	240
2.13.4 Tool Offset Amount Setting (G10)*	248
2.13.5 Progarm Call Function	250
2.13.5.1 Subprogarm (M98, M99)	250
2.13.6 Drilling Canned Cycle *	252
2.13.6.1 Drilling canned cycle *	252
2.14 MACRO PROGRAM (G65, G66, G67)*	267
2.14.1 Macro Progarm Call	269
2.14.1.1 Macro nesting call	274
2.14.1.2 Argument designation	276
2.14.2 Variables	280
2.14.3 Specifying Variables	297
2.14.4 Quoting Variables	298
2.14.5 Undefined Variables	299
2.14.6 Arithmetic Commands	300
2.14.7 Control Commands	303
2.14.8 Entering Macro Programs	308
2.14.9 RS-232C Data Output-2 (BPRNT, DPRNT)	309
2.14.10 Macro Program Alarm Numbers	314
2.14.11 Macro Program Examples	315
2.15 AUTO MEASUREMENT FUNCTION	322
2.15.1 Skip Function (G31)*	322
2.16 STORED STROKE LIMIT CHECK	324
2.16.1 Stored Stroke Limit A	324
2.16.2 Stored Stroke Limit B	325
2.17 MULTI-SELECTION SUPPORT FUNCTION (FOR MULTI-SECTION OPTION ONLY)	332
2.17.1 Section Synchronization Command *	332
2.17.2 Password Command	337
2.17.3 Control Axis Select Command	340
2.17.4 Local Section Spindle Selection Function *	341
2.17.5 Synchronous Feed Reference Spindle Select *	343
2.17.6 Remote Program M, T Command *	344
2.17.7 Pulse Copy Function*	345

2.1 CONTROL AXES

2.1.1 Control Axes and Number of Simultaneously Controllable Axes

Table 2.1.1 shows control axes and the number of simultaneously controllable axes.

Table 2.1.1 Control Axes and Number of Simultaneously Controllable Axes

Item			Explanation
Control axis	Single-program control	Basic control axis	X and Z axes
		Additional axis control A*	X, Z, and C axes
		Additional axis control B*	Up to five axes can be expanded (Y-axis, B-axis, etc.)
	Multi-program control*		Up to eight control axes and up to three programs can be expanded. [Maximum number of control axes: 5 axes per program] (Example of typical control axis) • Two-program, four-axis control: Program 1 X and Z axes, Program 2 X and Z axes • Two-program, six-axis control: Program 1 X, Z, and C axes, Program 2 X, Z, and C axes • Three-program, eight-axis control: Program 1 X, Z, C axes, Program 2 X, Z, C axes, Program 3 X and Z axes
Number of simultaneously controllable axes	Positioning (G00)		All axes simultaneously controllable
	Linear interpolation (G01)		All axes simultaneously controllable
	Circular interpolation (G02, G03)		Two axes simultaneously controllable **(NOTES)**
	Manual control		All axes simultaneously controllable **(NOTES)**

NOTES

1. During polar coordinate interpolation* or cylindrical interpolation*, circular interpolation can be performed on hypothetical XC plane and hypothetical ZC plane. For details, refer to Par. 2.4.5, "POLAR COORDINATE INTERPOLATION" and Par. 2.4.6, "CYLINDRICAL INTERPOLATION."
2. Manual control by manual pulse generator is basically simultaneously one-axis control.

2.1.2 Input Increment and Output Increment

2.1.2.1 Least input increment and least input increment × 10

(1) The minimum input units that can be instructed by punched tape or MDI are shown in Table 2.1.2.

Table 2.1.2 Least Input Increment

(pm1000 D0 - "0")

	Linear Axis (X, Y, Z, etc.)	C-Axis*
Metric Input	0.001 mm	0.001 deg
Inch Input	0.0001 in	0.001 deg

Least input increment times ten can be set as shown in **Table 2.1.3** by setting parameter pm1000 D_0 at "1."

Table 2.1.3 Input Increment × 10

(pm1000 D0 - "1")

	Linear Axis (X, Y, Z, etc.)	C-Axis*
Metric Input	0.01 mm	0.01 deg
Inch Input	0.001 in	0.01 deg

NOTE Metric input and inch input can be selected by setting pm0007 D_0 or G20/G21 command.

(2) Tool offset value must always be written in 0.001 mm (or 0.0001 inch, or 0.001 deg*) and offset is possible in these units.

(3) In 0.01 mm increment system, the following operation must be made in units of 0.01 mm.

(a) Write operation in MDI mode

(b) Programming for operation in MEMORY mode

(c) Program editing operation in EDIT mode

NOTES

1. If NC programs set by 0.001 mm are fed into or stored in equipment set by 0.01 mm increment, the machine will move ten times the intended dimensions.
2. If the increment system is switched when the contents of NC tape are stored in memory, the machine will move by either ten times or one-tenth of the instructed dimensions.
3. When the stored program is punched out on the tape, the stored figures are punched out "as stored" regardless of switching of the increment system.

2.1.2.2 Least output increment

Least output increment is the minimum unit of tool motion.

Table 2.1.4 Least Output Increment

	X-Axis (radius value)	Z-Axis	C-Axis*
Metric Output	0.0005 mm	0.001 mm	0.001 deg
Inch Output	0.00005 in	0.0001 in	

NOTE The least output increment of extra axes such as Y-axis and B-axis depends on the machine. Refer to the machine manufacturer's manual.

2.1.3 Maximum Move Command Values

(1) Maximum move command values are shown below. The values in **Table 2.1.5** apply not only to move command addresses X, Z, Y, C, U, W, V, and H, but also to distance command addresses I, K, J, R, A, and B.

Table 2.1.5 Maximum Move Command Values

		Linear Axis (X, Z, Y, etc.)	C-Axis*
Metric Output	Metric Input	±999999.999 mm	±999999.999 deg
	Inch Input	±39370.0787 in	±999999.999 deg
Inch Output	Metric Input	±999999.999 mm	±999999.999 deg
	Inch Input	±99999.9999 in	±999999.999 deg

NOTE If the maximum values are exceeded, correct operation is not guaranteed.
1. In incremental programming, input values must not exceed the maximum command value.
2. In absolute programming, move quantity of each axis must not exceed the maximum command value.

The cumulative value must not exceed the values shown in **Table 2.1.6**.

Table 2.1.6 Maximum Cumulative Values

	Linear Axis (X, Z, Y, etc.)	C-Axis*
Metric input	±999999.999 mm	±999999.999 deg
Inch input	±99999.9999 in	±999999.999 deg

NOTE Listed input values do not depend on metric/inch output system.

2.2 PROGRAMMING

2.2.1 Process Sheet

(1) Programs are first drafted on process sheets.

(2) Process sheets should be easy to read and to make corrections, and should be designed and prepared by the user in conformity with the specifications of the NC.

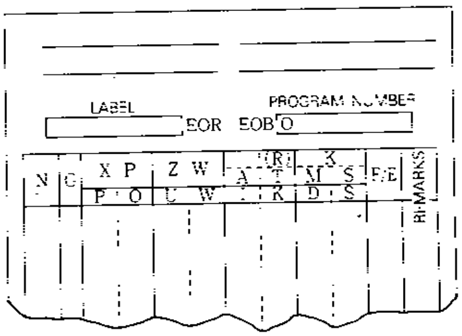
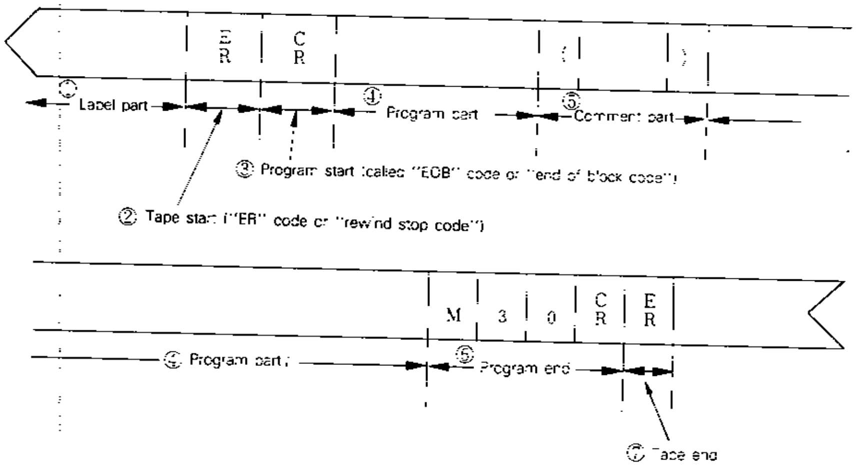


Fig. 2.2.1 Example of Process Sheet

2.2.2 Programming Format

The general format of a part program is as shown below.

① Label (See Par. 2.2.2.1.)
② Tape start (See Par. 2.2.2.2.)
③ Program start (See Par. 2.2.2.2.)
④ Program Part (See Par. 2.2.2.4.)
⑤ Comment Part (See Par. 2.2.2.5.)
⑥ Program end (See Par. 2.2.2.3.)
⑦ Tape end (See Par. 2.2.2.2.)



NOTE As the program ends, M02 and M99 can be substituted for M30.

Fig. 2.2.2 Tape with Single Program (EIA code)

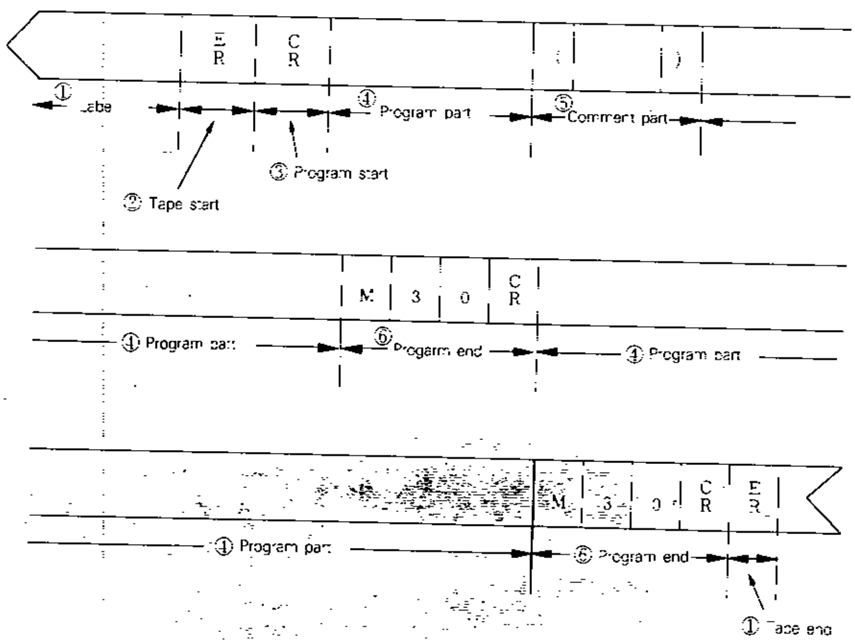


Fig. 2.2.3 Tape with Multi-Program (EIA Code)

2.2.2.1 Label part and label skip

To facilitate tape classification or handling, an arbitrary "label" may be written in the first part of punched tape. The label skip function allows a section of data up to the first EOB code to be skipped, so that off-specification addresses or function codes can be used. In addition, parity-disregarded, modified code can also be used.

The label skip function is valid in the following cases in which case "LSK" is indicated on the CRT screen.

① When the power is turned on
② When the RESET key on the CRT operation panel is used
③ When BG reset operation is performed

When label skip is valid, information up to the first EOB code on punched tape is disregarded.

2.2.2.2 Tape start/tape end

Punch the same code in tape start and tape end. See **Table 2.2.1**.

Table 2.2.1 Tape Start/Tape End

EIA	ISO	Meaning
ER	%	Tape start/Tape end

(1) The ER code (rewind stop code) following the tape start label indicates the stop point when tape is rewound by a tape rewind command.

(2) The ER code for tape stop indicates the stop point when multiple part programs are stored in the NC memory.

2.2.2.3 Program start / progarm end

(1) Program start

The following code is punched to declare the beginning of the program section. This is required to clear the label skip.

Table 2.2.2 Program Start

EIA	ISO	Meaning
CR	LF/NL	Program start

(2) Program end

The following code is punched to declare the end of the program section.

Table 2.2.3 Program End

EIA	ISO	Meaning
M02CR	M02LF/NL	Program end
M30CR	M30LF/NL	Program end & rewind
M99CR	M99LF/NL	Sub-program end

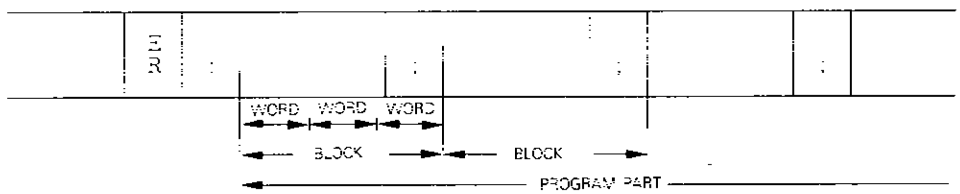
NOTES
1. When M02CR, M02LF/NL, M30CR or M30LF/NL is executed, the equipment may or may not be reset or rewound depending on equipment specifications. Refer to the command manuals issued by your machine tool manufacturer.
2. When multiple part programs are stored in the NC memory, control may move to the next after reading the program end code shown above. This occurs when part programs are entered by total input.
3. If M02 or M30 is not present at the end of the program section and ER or LF/NL is immediately executed, the NC machine may be reset.

2.2.2.4 Program part

A punched section from the program start all the way to the program end is called the program section.

The program section consists of several blocks, and each block is comprised of several words.

Each block is separated by the EOB code (;).



NOTE EOB (program start) is represented by ";" in this manual.

Fig. 2.2.3 Block Definition

(1) Program number

(a) By adding a program No. immediately behind the program start code, it is possible to discriminate your program from other programs.

(b) A program No. can be specified by up to five digits of numbers following address 0. Up to 99 program Nos. can be entered for the equipment. However, the maximum number of program Nos. can be increased to 299 or 999 by adding options.

(2) Sequence number

(a) Integers consisting of up to 5 digits may be written following an address character N as sequence numbers.

(b) Sequence numbers are reference numbers for blocks, and do not have any influence on the meaning or sequence of machining processes. Therefore, they may be sequential, non-sequential, or duplicated numbers.

NOTE Generally, sequential numbers are convenient as sequence numbers.

(c) When searching for sequence numbers, be sure to search or specify program numbers beforehand.

NOTES

1. When 6 or more digits are written as a sequence number, only the digits up to the 5th from the trailing end are effective.
2. When two or more blocks have the same sequence number, only one is retrieved and read, and no more searching is performed.
3. Blocks without sequence numbers can also be searched for with respect to the address data contained in the blocks.

(3) Word

A block is a collection of words. A word consists of an address (alphabet) and a numeral of several digits that follow the address.

G	02		Z	−10.	
Address	Numeral		Address	Numeral	(Includes a sign and decimal point)
Word			Word		

2.2 PROGRAMMING (Cont'd)

(a) Address characters and their meanings

Table 2.2.4 Address Characters

B: Basic
O: Option

Address Characters	Meanings	
A	G01 and 111 angle specifications, G76 thread angle	O
B	Spindle dislocation angle of multiple-thread screws. Compound chamfering, Rounding angle specification	O
C	C-axis coordinate value	O
D	C71 to C76 cutting quantity, Cutting count	O
E	Precise feed specification, Screw cutting precise lead specification	B
F	Normal feed specification. Screw cutting normal lead specification	B
G	Preparatory function	B
H	C-axis incremental command value	O
I	X-axis component of circular arc center, Parameter of canned cycle, Chamfering quantity (radius value)	B, O
J	Y-axis component of circular arc center	O
K	Z-axis component of circular arc center. Parameter of canned cycle, Chamfering quantity	B, O
	Incremental/reduced quantity of variable lead screw cutting	O
L	Repetition count	B, O
M	Miscellaneous function (M function)	B
N	Sequence number	B
O	Program number	B
P	Dwell, Starting sequence number of canned cycle, Program number, Macroprogram number	B, O
Q	Starting sequence number of subprogram. Ending sequence number of canned cycle	B, O
	Cutting quantity of drilling canned cycle	O
R	Circular arc radius. Rounding quantity. Nose radius quantity. R point coordinates of drilling canned cycle	B, O
S	Spindle function (S function), Spindle maximum rotation speed	B
T	Tool function (T function), Tool coordinate memory number	B, O
U	X-axis incremental command value, Dwell, Parameter of canned cycle	B, O
V	Y-axis incremental command value	O
W	Z-axis incremental command, Parameter of canned cycle	B, O
X	X-axis coordinate value	B
Y	Y-axis coordinate value	O
Z	Z-axis coordinate value	B

(b) Function characters and their meanings

Table 2.2.5 Function Characters

EIA Code	ISO code	Meaninng	Remarks
Blank	NuL	Fault in significant data area in EIA, Disregarded in ISO	
BS	BS	Disregarded	
Tab	HT	Disregarded	
CR	LF/NL	End of Block (EOB)	
SP	CR	Disregarded	
	SP	Space	
ER	%	Rewind stop	
UC		Upper shift	
LC		Lower shift	
2-4-5 bits	(	Control out (Comment start)	EIA: Special code
2-4-7 bits	)	Control in (Comment end)	
+	+	Disregarded, User macro operator	
−	−	Minus sign, User macro operator	
0 to 9	0 to 9	Numerals	
a to z	A to Z	Address characters	
/	/	Optional block skip, User macro operator	
Del	DEL	Disregarded (Including All Mark)	
•	•	Decimal point	
Parameter setting	#	Sharp (Variable)	
*	*	Asterisk (Multiplication operator)	
=	=	Equal mark	
[	[	Left bracket	
]	]	Right bracket	EIA: Special code
O	:	For comment in Microprogram	
$	$	For comment in Microprogram	
@	@	For comment in Microprogram	
?	?	For comment in Microprogram	
,	,	For comment in Microprogram	

NOTES
1. Characters other than the above cause a fault in significant data area.
2. Information between Control Out and Control In is disregarded as insignificant data.
3. Tape code (EIA or ISO) can be automatically identified during input; during output, it can be designated by setting pm0004 D6.

(c) Numeral specification range (input format)

(i) A variable block format conforming to JIS (Japanese Industrial Standard) B6313 is used for YASNAC i80L.

(ii) **Table 2.2.6** shows numeral specification ranges. Numerals following the address characters in Table 2.2.6 indicate the programmable number of digits.

A decimal point should be omitted in actual programming. Processing is performed differently, depending on whether a decimal point is specified. For details, see Par. 2.8.3.

Explanation of input format

X + 6 3

- 3 — Down to third decimal place
- 6 — Six digits of integer
- + — Sign
- X — Address character: X

(Down to third decimal place and Six digits of integer: Differs in metric and inch input format. [See (e) and (f).])

(iii) The leading zeros and the plus sign (+) can be omitted for all address codes including sequence numbers.

NOTE The minus sign (−) cannot be omitted.

(iv) In the manual, EOB (end of block) code in a program example is represented by a semicolon (;). In actual programming, CR (EIA code) or LF/NL (ISO code) should be used instead of the semicolon (;).

(v) Metric input format

O5 N5 G3 a+63

F33 S5 T(2+2) M3;

(vi) Inch input format

O5 N5 G3 a+54

F24 S5 T(2+2) M3;

NOTES 1. "a" represents axis command addresses such as X, Z, Y, C, U, W, V, H, I, and K.
2. P, Q, R and L are omitted in the above format because they are used for various meanings such as canned cycles.

Table 2.2.6 Input Formats

No.	Address	Metric Output		Inch Input		B: Basic O: Option
		Metric input	Inch Input	Metric input	Inch input	
1	Program No.	O5		O5		B
2	Sequence No.	N5		N5		B
3	G function	G3		G3		B
4	Coordinate Address a: X,Z,I,K,U W,R, Q,Y,J	a + 63	a − 54	a + 63	a + 54	B, O
	b: C,H	b + 63		b + 63		O
5	Feed per minute	F60 or F63	F51 or F54	F60 or F63	F51 or F54	B
6	Feed and screw lead per rotation	F33	F24	F33	F24	B
		E34	E26	E34	E26	B
7	S-function	S5		S5		B
8	T-function	T (2+2)		T (2+2)		B
		T (3+3)		T (3+3)		O
9	M-function	M3		M3		B
10	Dwell	U (P) 63		U (P) 63		B
11	Program No. designation	P5		P5		B
12	Sequence No. designation	Q (P) 5		Q (P) 5		B, O
13	No. of repetitions	L9		L9		B
14	Line angle designation	A (B) 33		A (B) 33		O
15	Multiple-thread screw angle designation	B3		B3		O

NOTE Input format for feeding per minute is selected by pm2004 D0.

pm2004 D0=0 →F60 mm/min, F52 in/mm
pm2004 D0=1 →F63 mm/min, F54 in/mm

(4) Block

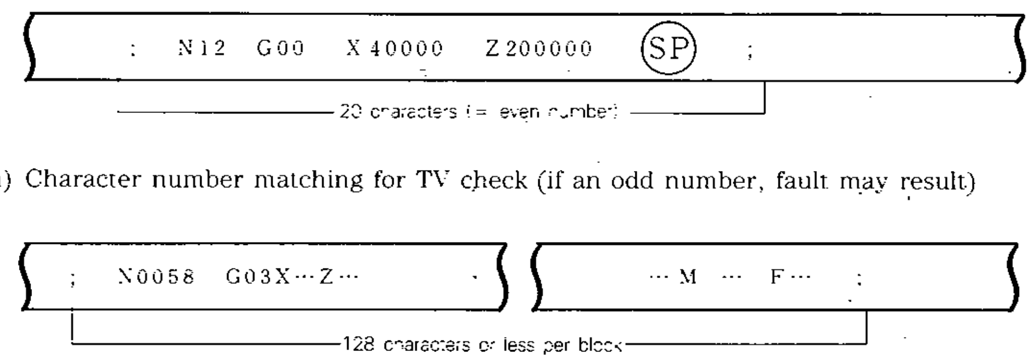
(a) One block is terminated with the end of block (EOB) code. The EOB code is represented by "CR" in EIA or "LF/NL" in ISO.

NOTE The EOB code in programming examples in this manual is indicated by ";" for simplicity.

(b) Use only the address codes and function codes shown in Tables 2.2.4 and 2.2.5.

(c) Up to 128 characters can be written in one block.

NOTE The maximum number of characters does not include Del and other invalid characters.



(a) Character number matching for TV check (if an odd number, fault may result)

(b) Number of effective characters allowed in one block

Fig. 2.2.5 Block

(5) Optional block skip (/1) (/2 to /9)*

Those blocks in which "/n" (n = 1 to 9) is included are disregarded between /n and the end of that block, when the external optional block skip switch for that number "n" is on; /n can be specified in the middle of a block.

NOTE With "/1", "1" can be omitted.

(EXAMPLE) | /2 N1234 G00 X100/3 Z200; |

When the switch for /2 is on, the entire block is disregarded, and when the switch for /3 is on, this block is read as if

| N 1234 G01 X100; |

NOTES

1. The optional block skipping process is executed while a part program is read from the tape into the buffer storage. If the program has already been read, subsequent switching on is ineffective to skip the blocks.
2. While reading or punching out programs, the skip function is ineffective.

2.2.2.5 Comment part

Message display by control-out/control-in

(1) **Control-out/control-in programming and display:**

Any desired message can be programmed in a section enclosed with the control-out and control-in codes in a part program for display on the CRT screen.

In this case, the information enclosed between the control-out and control-in codes is assumed to be meaningless information.

(2) **Method of editing control-out and control-in codes:**

The control-out and control-in section can be edited by normal edit operation.

Depress SHIFT key, then depress U key to input character "(."

Depress SHIFT key, then depress V key to input character ")."

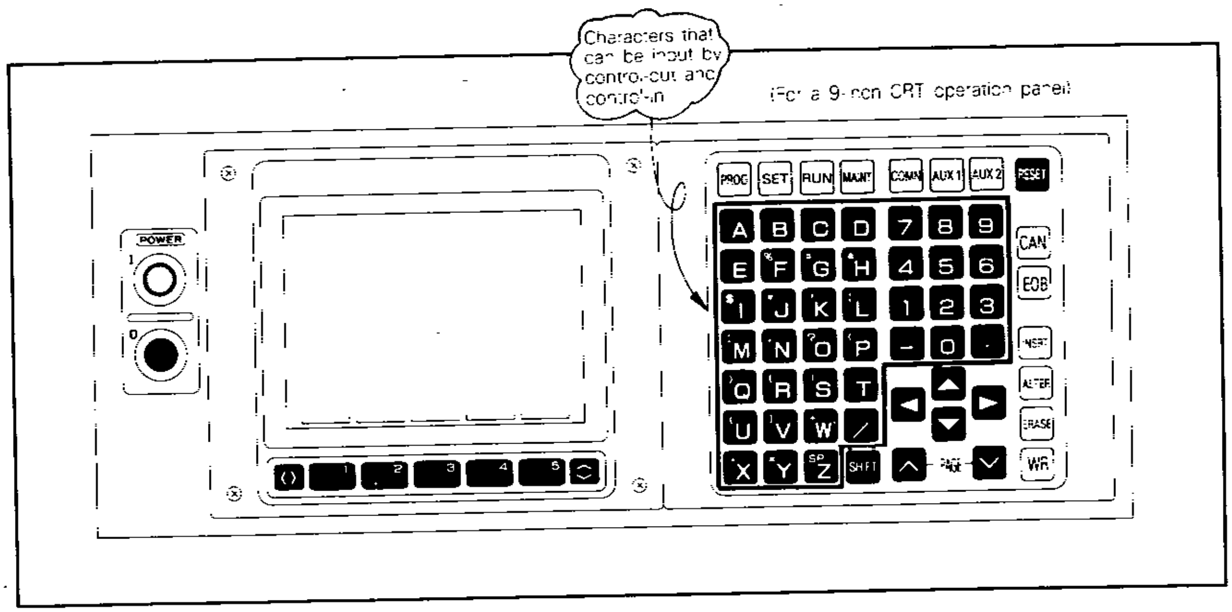


Fig. 2.2.6 Characters That Can Be Entered by Control-out and Control-in (Keys Enclosed by Line)

NOTES
1. Characters can be entered between control-out and control-in by using the keys that are enclosed by line in Fig. 2.2.6.
2. Control-out and control-in cannot be entered in a section already enclosed by control-out and control-in.

(Example of programming)

```
        ⋮
(TEST PROGRAM);
G00 X100. Z100. ;
G01 X0  Z0  F10.;
(DRILL END);
        ⋮
```

(Example of message display by control-out or-in)

```
PROGRAM RUN                         O12345 N00018
        ⋮
(TEST PROGRAM);
G00 X100. Z100.;
G01 X0 Z0 F10.;
(DRILL END);
        ⋮
```

2.2.3 Buffer Register and Multi-active Register

(1) Buffer register

(a) During normal operation, two blocks of data are read in advance and offset computing is made for the following operation.

(b) In the nose radius compensation, two blocks of data (or up to 4 blocks of data) are read in advance and compensation computing required for the next operation is executed.

(c) One bclock can contain up to 128 characters including EOB.

(2) Multi-active Register*

(a) For the portion of a part program sandwiched in between M93 and M92, up to 7 blocks of data are read in advance.

Table 2.2.7 M93 and M92 Codes

M code	Meaning
M93	Multi-active register off
M92	Multi-active register on

(b) Inter-block stoppage can be eliminated when the program is so made that the automatic operation time of advance reading of 7 blocks is longer than the processing time of advance reading of the next 7 blocks of data.

NOTE Advance reading is not made for every 7 blocks but is always ready to be made up to 7 blocks in M93 mode.

2.2.4 Tape Code

2.2.4.1 Tape codes

(1) With this control, both the EIA codes and ISO codes can be used.

EIA code .... EIA RS-244-A

ISO code .... ISO 840

Table 2.2.8 shows the EIA and ISO punched tape formats.

(2) Before starting to program any machining operation, a decision must be made as to the code to be used.

2.2.4.2 EIA/ISO auto-select

(1) The equipment code must be switched over prior to operation by the completed NC tape.

(2) The code is automatically discriminated regardless of the contents of setting No. pm0004 D0. This is automatically set so that the code is determined by the EOB code first read in a lable skip state and the subsequent data are read by the determined code.

(3) For punching tapes, the code must be selected by the setting of pm0004 D0.

When the contents of setting number pm0004 D0 are "1" ... EIA code

When the contents of setting number pm0004 D0 are "0" ... ISO code

Table 2.2.8 Tape Code

EIA Code 8	7	6	5	4	o	3	2	1	Characters (EIA)	Characters (ISO)	ISO Code 8	7	6	5	4	o	3	2	1
		O			o				0	0			O	O		o			
					o			O	1	1	O		O	O		o			O
					o		O		2	2	O		O	O		o		O	
			O		o		O	O	3	3			O	O		o		O	O
					o	O			4	4	O		O	O		o	O		
			O		o	O		O	5	5			O	O		o	O		O
			O		o	O	O		6	6			O	O		o	O	O	
					o	O	O	O	7	7	O		O	O		o	O	O	O
					o				8	8	O		O	O	O	o			
			O	O	o			O	9	9			O	O	O	o			O
	O	O			o			O	a	A		O				o			O
	O	O			o		O		b	B		O				o		O	
	O	O	O		o		O	O	c	C	O	O				o		O	O
	O	O			o	O			d	D		O				o	O		
	O	O	O		o	O		O	e	E	O	O				o	O		O
	O	O	O		o	O	O		f	F	O	O				o	O	O	
	O	O			o	O	O	O	g	G		O				o	O	O	O
	O	O			o				h	H		O			O	o			
	O	O	O		o			O	i	I	O	O			O	o			O
	O		O		o	O			j	J	O	O			O	o		O	
	O		O	O	o		O		k	K		O			O	o		O	O
	O			O	o		O	O	l	L	O	O			O	o	O		
	O		O		o	O			m	M		O			O	o	O		O
	O				o	O		O	n	N		O			O	o	O	O	
	O				o	O	O		o	O	O	O			O	o	O	O	O
	O		O		o	O	O	O	p	P		O		O		o			
	O		O	O	o				q	Q	O	O		O		o			O
	O			O	o			O	r	R	O	O		O		o		O	
		O	O		o		O		s	S		O		O		o		O	O
		O			o		O	O	t	T	O	O		O		o	O		
		O	O		o	O			u	U		O		O		o	O		O
		O			o	O		O	v	V		O		O		o	O	O	
		O			o	O	O		w	W	O	O		O		o	O	O	O
		O	O		o	O	O	O	x	X	O	O		O	O	o			
		O	O	O	o				y	Y		O		O	O	o			O
		O		O	o			O	z	Z		O		O	O	o		O	
					o				Blank	NUL						o			
		O		O	o		O		BS	BS	O				O	o			
		O	O	O	o	O	O		Tab	HT					O	o			O
O					o				CR	LF/NL					O	o		O	
					o				—	CR	O				O	o	O		O
			O		o				SP	SP	O		O			o			
				O	o		O	O	ER	%	O		O			o	O		O
	O	O	O	O	o	O			UC	—						o			
	O	O	O	O	o		O		LC	—						o			
			O	O	o		O		—	(			O		O	o			
	O			O	o		O		—	)	O		O		O	o			O
	O	O	O		o				+	+			O		O	o		O	O
	O				o				-	-			O		O	o	O		O
	O				o	O	O		:	:			O	O	O	o		O	
		O	O		o			O	/	/	O		O		O	o	O	O	O
	O	O	O	O	o	O	O	O	Del	DEL	O	O	O	O	O	o	O	O	O
O	O	O	O	O	o	O	O	O	All Mark	All Mark	O	O	O	O	O	o	O	O	O
See Note 2.									#	#	O		O			o		O	O
O			O	O	o				*	*	O		O		O	o		O	
O				O	o	O			=	=	O		O	O	O	o	O		O
O		O	O		o				[	[	O	O		O	O	o		O	O
O		O			o		O		]	]	O	O		O	O	o	O		O
O			O		o	O			$	$			O			o	O		
O				O	o	O	O	O	@	@	O	O				o			
O			O	O	o	O	O		?	?			O	O	O	o	O	O	O
	O	O		O	o		O	O	•	•			O		O	o	O	O	

NOTES
1. For characters from # to ?, EIA codes have not been agreed upon. In the present system, for the time being, the provisional codes on the left are used.
2. EIA code of character # can be designated by the parameter pm4100.

2.2.5 NC Tape

2.2.5.1 Paper tape

(1) For part program tapes, eight-channel paper tapes for computers conforming to JIS C6243 (width: 25.4 ± 0.08 mm, thickness: 0.108 mm) are used.

(2) The color should be black or gray. Tapes with high transparency tend to cause reading errors, and should not be used.

2.2.5.2 NC tape punch

(1) Part programs written on process sheets are punched in EIA or ISO codes in paper tape with a tape puncher.

(2) A part program tape should be provided with a proper length of feed holes at the leading and the trailing ends.

NOTE For a tape reader using 6" reels*, the feed hole should be at least 70 cm, and for a tape reader using 8" reels, it should be at least 1 m.

2.2.5.3 NC tape check

Punched part program tapes can be checked by an NC with the following functions.

- Machine lock
- M function lock
- Dry run
- Single-block operation

2.2.5.4 Splicing NC tape

To join (splice) part program tapes, the two ends should be placed end to end without overlapping and without a space. A proper length of splicing tape (approx. 0.08 mm in thickness) should be applied on one side. Tape splices are available in the fully perforated type and in the type in which only the feed holes are punched, but the former is more convenient.

After splicing, the tape should be checked for correct alignment of the feed holes before use.

NOTE Do not use rigid industrial adhesive, and do not make the joint too thick, as these conditions are conducive to jamming troubles.

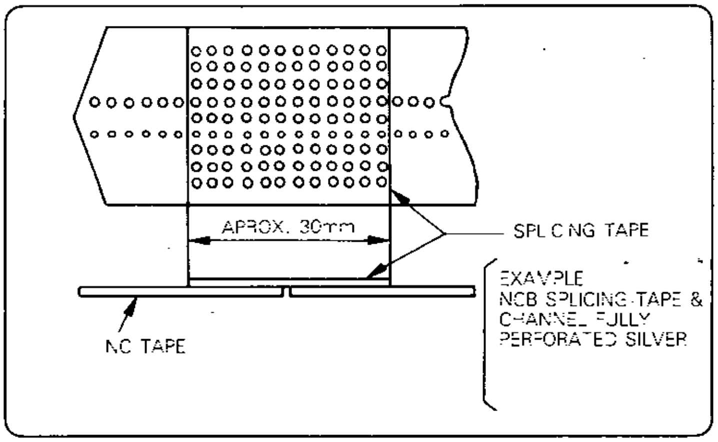


Fig. 2.2.7 Splicing Part Program Tape

2.2.5.5 Keeping NC tape

Generally, properly maintained part program tapes can last at least 300 cycles, with one cycle consisting of one reading and one rewinding pass.

NOTES
1. Part program tapes should be stored in a clean area, free of contaminants and humidity.
2. Do not handle part program tapes wearing gloves contaminated with oil or cutting fluid.

2.3 PREPARATORY FUNCTION (G-FUNCTION)

All G codes in this manual are explained with standard code.

An address character G and up to 3 digits following it specify the operation of the block.

(1) Normal G codes are classified as non-modal G codes and modal G codes. The codes belonging to division B are included in the basic specificaions.

Non-modal G codes (marked with *)

Effective only for specified blocks.

Modal G codes (groups 01 to 31)

Once these codes are instructed, they are effective until another G code in the same group is instructed.

They can be programmed twice or more in the same block.

NOTE When different G codes in the same group are programmed, the last appearing G code is effective.

(2) Special G code I can be substituted for standard G code by setting parameter pm400 D7 = 1.

(3) When special G code II is selected as an option, the special G code II becomes effective by setting parameter pm400 D7 to 1. (If the option is not selected, special G code I becomes effective by setting the same parameter.) Setting pm400 D7 to 0 makes standard G code effective.

(4) G code shown in **Table 2.3.1** can be selected by parameter setting.

(a) The following G codes can select the G code state at power on.

Table 2.3.1 Selection of G Code State at Power ON

Group	G Code to be Applied	Parameter	Bit	Code
01	G00 or G01	pm4000 D0	0	G00
			1	G01
03	G90 or G91	pm4000 D1	0	G90
			1	G91
04	G98 or G99	pm4000 D2	0	G98
			1	G99

(5) The preset of external current value display can be selected at G50 execution.

pm3000 D0

Bit	Meaning
0	Not preset
1	Preset

(6) At reset, G code belonging to group 01 and 03 can be set by parameter setting.

Table 2.3.2 Setting of G Code at Reset

Group	Timing	Parameter pm4000 D0 = 1	Parameter pm4000 D0 = 0
01	At power ON	G01	G00
	At reset	G01	G00
03		Parameter pm4000 D6 = 1	Parameter pm4000 D6 = 0
	At power ON	pm4000 D1 = 0 .... 90 pm4000 D1 = 1 91	pm4000 D1 = 0 .... 90 pm4000 D1 = 1 91
	At reset	Immediately preceding G code is stored.	pm4000 D1 = 0 .... 90 pm4000 D1 = 1 91

(7) The status of G code of groups 05, 07, and 08 depends on the respective corresponding setting data (pm0007D0, pm0008D0, and pm0008D1).

(8) Circular radius specification can be instructed by G code of each of G02/03 and G22/23.

(9) Chamfering and rounding commands can use G01 instead of G11/12.

B: Basic
O: Option

Table 2.3.3 G Codes

Standard G code	Special G code I	Special G code II	Group	Function		Division
G00	G00	G00	01	Positioning (rapid feed)		B
G01	G01	G01		Linear interpolation, Angle specification line interpolation		B
G02	G02	G02		Circular interpolation CW (radius R specification)		B
G03	G03	G03		Circular interpolation CCW (radius R specification)		B
G04	G04	G04	*	Dwell		B
G06	G06	G06		Positioning in fault detect off mode		B
G10	G10	G10		Tool offset quantity setting		O
G11	G11	G11	01	Chamfering		B
G12	G12	G12		Rounding		B
G17	G17	G17	16	XY plane designation		O
G18	G18	G18		ZX plane designation		O
G19	G19	G19		YZ plane designation		O
G20	G20	G70	05	Inch input designation		B
G21	G21	G71		Metric input designation		B
G22	G22	G22	01	Circular interpolation radius designation CW		B
G23	G23	G23		Circular interpolation radius designation CCW		B
G27	G27	G27	*	Reference point return check		B
G28	G28	G28		Automatic return to reference point		B
G29	G29	G29		Return from reference point		B
G30	G30	G30		Return to 2nd, 3rd, 4th reference points		B, O
G31	G31	G31		Skip function		B
G32	G33	G33	01	Screw cutting, Continuous screw cutting.		B, O
G34	G34	G34		Multiple thread screw cutting		O
G36	G36	G36	07	Stored stroke limit 2nd area off		O
G37	G37	G37		Stored stroke limit 2nd area on		O
G38	G38	G38	08	Stored stroke limit 3rd area off		O
G39	G39	G39		Stored stroke limit 3rd area on		O
G40	G40	G40	06	Nose radius compensation cancel		O
G41	G41	G41		Left nose radius compensation		O
G42	G42	G42		Right nose radius compensation		O
G50	G92	G92	*	Coordinate system setting. Spindle maximum rotation speed setting. Work coordinate system setting.		B, O
G51	G51	G51		Return to current value display origin		O
G65	G65	G65	*	Simple call of macroprogram		O
G66	G66	G66	09	Modal call of macroprogram		O
G67	G67	G67		Canceling of modal call of macroprogram		O
G70	G70	G72	*	Compound canned cycle	Finishing cycle	O
G71	G71	G73			Outer diameter rough cutting cycle	O
G72	G72	G74			End face rough cutting cycle	O
G73	G73	G75			Closed loop cutting cycle	O
G74	G74	G76			End face cutting-off cycle	O
G75	G75	G77			Outer diameter cutting-off cycle	O
G76	G76	G78			Automatic screw cutting cycle	O

NOTES 1. The codes maked with are automatically selected at power on or reset.
2. The codes marked with are automatically selected at power on.

B: Basic
O: Option

Table 2.3.3 G Codes (Cont'd)

Standard G code	Special G code I	Special G code II	Group	Function		Division
G80	G80	G80		Drilling fixed cycle	Canceling of drilling canned cycle	O
G81	G81	G81			Drill cycle	O
G82	G82	G82			Spot facing cycle	O
G83	G83	G83			Deep hole drill cycle	O
G831	G831	G831			Fast deep hole drill cycle	O
G84	G84	G84			Tap cycle	O
G841	G841	G841	14		Reverse tap cycle	O
G85	G85	G85			Boring cycle	O
G86	G86	G86			Boring cycle	O
G861	G861	G861			Boring cycle	O
G87	G87	G87			Back boring cycle	O
G88	G88	G88			Boring cycle	O
G89	G89	G89			Boring cycle	O
G90	G77	G20	01	Cutting cycle A		B
G92	G78	G21		Screw cutting cycle		B
G94	G79	G24		Cutting cycle B		B
G93	G93	G93	04	Solid tap mode		O
G96	G96	G96	02	Peripheral speed keeping control		O
G97	G97	G97		Canceling of peripheral speed keeping control		O
G98	G94	G94	04	Feed per minute (mm/min)		B
G99	G95	G95		Feed per revolution (mm/rev)		B
—	G90	G90	03	Absolute command		B
—	G91	G91		Incremental command		B
G105	G105	G105	*	Rotation tool spindle synchronization, Rotation ratio setting		O
G111	G111	G111		Taper compound chamfering and rounding		O
G112	G112	G112		Arc compound chamfering and rounding		O
G122	G122	G122	11	Tool register start		O
G123	G123	G123		Tool register end		O
G124	G124	G124	20	Cylindrical interpolation mode on		O
G125	G125	G125		Cylindrical interpolation mode off		O
G126	G126	G126	19	Polar coordidnate mode on		O
G127	G127	G127		Polar coordinate mode off		O
G130	G130	G130	21	Control axis selection command		O
G131	G131	G131		Control axis selection command cancel		O
G132	G132	G132	22	Rotation tool S command mode		O
G133	G133	G133		Rotation tool S command mode cancel		O
G134	G134	G134	*	MT command to other systems		O
G136	G136	G136	23	Synchronized feeding spindle selection command		
G137	G137	G137		Synchronized feeding spindle selectin command cancel		
G198	G198	G198	15	Return to initial point of drilling canned cycle		O
G199	G199	G199		Return to radius point of drilling canned cycle		O

NOTES 1. The codes marked with ◥ are automatically selected at power on or reset.
2. The codes marked with ◹ are automatically selected at power on.

2.4 INTERPOLATION

2.4.1 Positioning (G00, G06)

The positioning is instructed by G00 and G06. The instructed axis moves at rapid speed to the position as follows.

- For absolute command: to the position on work coordinate system.
- For incremental command: from current position to the specified position

2.4.1.1 G00 (modal G code of 01 group)

The G00 command performs positioning in the fault detect on mode. This mode means that the program advances to the next block only after the servo lag pulses are decreased below the permissible level, and this is detected by the control. Therefore, with this command, the corner of the workpiece is machined sharply.

G00 X(U)....(W)....(*C(H).... *Y(V)....);

With this command, the tool is sent to the specified position in rapid traverse motions along the two axes (all axes* simultaneously). If any of the coordinate positions is not specified, the machine does not move along that coordinate axis.
Motions in the respective axis directions are independent of each other.

The rapid traverse rate for the respective axes are inherent to the machine tool. Refer to the machine tool builder's manual.

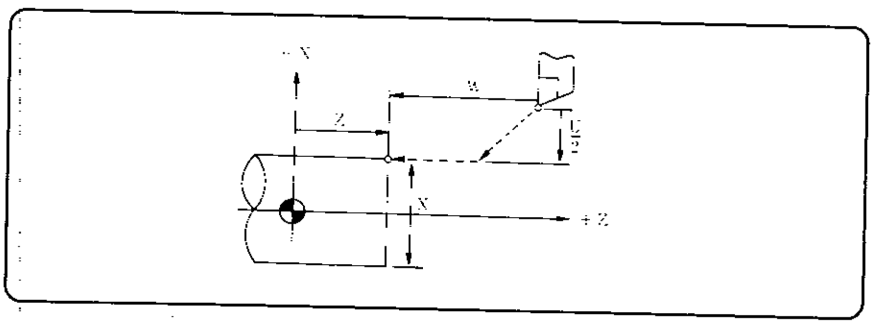


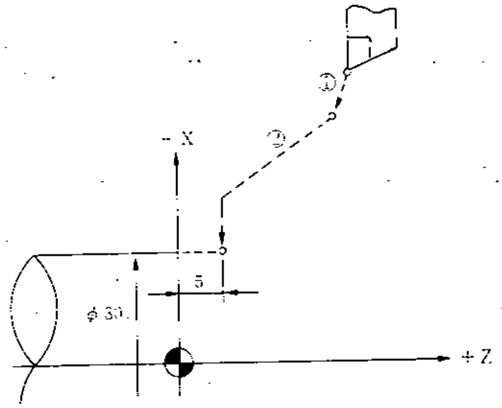
Fig. 2.4.1 Positioning

NOTES ON G00:

1. When G00 is instructed, motions in the respective axis directions are independent of each other, and therefore, the resultant tool path is not necessarily straight.
   When programming tool positioning commands, take care to avoid the possibility of tool and workpiece interference.
2. As a rule, when T code is instructed, issue G00 to the block. This is necessary to specify a movement speed of tool offset by the T code.

Example of programming

```
   G50 X150. Z100. ;
①  G00 T0101 S1000 M03 ;   ← Specifies offset movement speed.
   .
②  (G00) X30. Z5. ;        ← G00 may be omitted since it is a modal G code.
```



2.4.1.2 G06 (non-modal G code of ✻ group)

The G06 command performs positioning in the error detect off mode.

In this mode, the program advances to the next block after termination of pulse distribution. Therefore, the corner of the workpiece is machined round.

Use the following command:

```
G06 X(U)....Z(W)....(✻C(H)....✻Y(V)....);
```

2.4.2 Linear Interpolation (G01)

Linear interpolation can be performed by G01 command.

G01 X(U)....(W)....(∗C(H)....∗Y(V)....)F(E)....;

With this command, the tool is moved simultaneously in the two axial directions resulting in a linear motion.

(1) Feedrate

Specify by an F code or E code. In this case, the feedrate is controlled so that the instructed composite speed of all axes (tangential speed for component axial directions) becomes the specified value.

$$F = \sqrt{Fx^2 + Fz^2 + (Fc^2)}$$

(where Fx indicates feedrate in the X direction.)

NOTE If no F code or E code is given in the block containing the G01 or in preceding blocks, the block causes a "370" fault.

(2) F code

F code is instructed with feed quantity per rotation of spindle (mm per rotation or inch per rotation) or feed quantity (mm per minute or inch per minute) per minute.

(3) End point

The end point can be programmed either in absolute coordinates or in incremental values with addresses or G90 or G91. (Refer to Par. 2.8.1 "Absolute/Incremental Specification".)

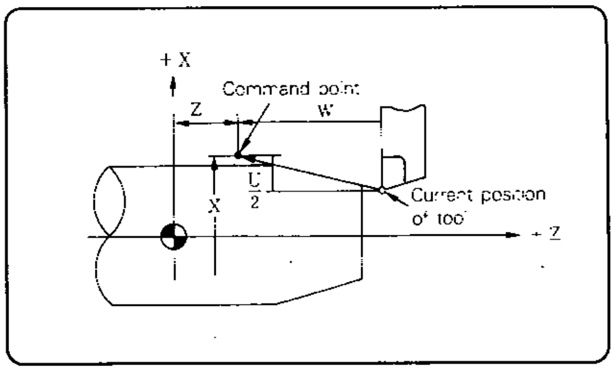
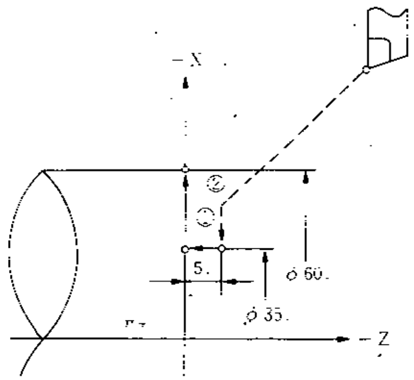


Fig. 2.4.2 Linear Interpolation

Example of Programming

```
G50  X100.   Z60. ;
G00  T0202   S600   M03 ;
     X35.    Z5. ;
G01  Z0      F1. ;
     X60.    F0.2 ;
```

①The tool is moved along a line resulting from linear interpolation specified by G01.



(4) When C-axis* is provided

When C-axis is provided optionally, for the same F code, the feedrates of X, Z and C axes are as indicated in **Table 2.4.1**.

Table 2.4.1 Feedrates of X, Z, and C Axes

F function (feed per minute)				In minimum F command unit	
				Feedrate of X and Z axes	Feedrate of C-axes
pm2004 D0=0	Metric output	Metric input	F60	1 mm/min	1 deg/min
		Inch input	F51	0.1 in/min	2.54 deg/min
	Inch output	Metric input	F60	1 mm/min	0.3937 deg/min
		Inch input	F51	0.1 in/min	0.1 deg/min
pm2004 D0=1	Metric output	Metric input	F63	0.001 in/min	0.001 deg/min
		Inch input	F54	0.0001 in/min	0.00254 deg/min
	Inch output	Metric input	F63	0.001 mm/min	0.0003937 deg/min
		Inch input	F54	0.0001 in/min	0.0001 deg/min

NOTE The C-axis cannot be used in the feed-per-rotation mode.

(5) Angle designation line interpolation *

The equipment having this option can instruct linear interpolation with an angle specified.

G01 X(U).....A.....F(E)....;

or

G01 Z(W).....A.....F(E)....;

This command performs linear interpolation with angle A from −Z-axis and single-axis distance in X- or Z-axis direction specified. The F or E code specifies the speed in tangential direction.

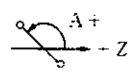
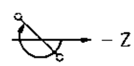
Table 2.4.2 shows the range of angles specified by address A.

Table 2.4.2 Range of Angle A

	Range of angle A
Metric input	0 to ±360.000°
Inch input	

Table 2.4.3 shows the meanings of signs of angle designation.

Table 2.4.3 Meanings of Signs of Angle Designation

Sign	Meaning	
A+	Angle Form −Z-axis in counterclockwise direction	
A−	Angle from +Z-axis in clockwise direction	

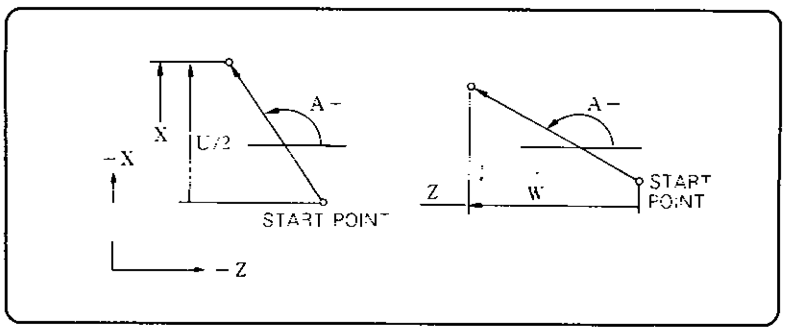
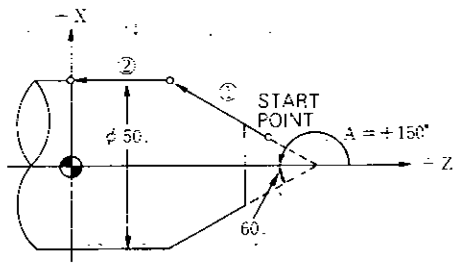


Fig. 2.4.3 Linear Interpolation with an Angle Specified

Example of Programming

```
:
G01  X50.  A  150.  F0.3 ; ←── ①
G01  Z0.   A−180. ;       ←── ②
```



(6) Chamfering, corner radius designation ✻

Chamfering is performed by the following command:

G01	{ X(U).....K.... / Z(W).....I..... }	F(E).....;

Rounding is perfomed by the following command:

G01	{ X(U).....R..... / Z(W).....R.... }	F(E).....;

For details, see Par. 2.4.3.

2.4.3 Chamfering Corner Radius Designation (G11, G12) *

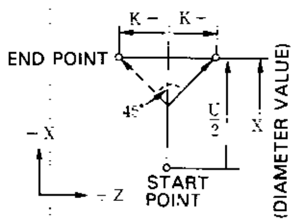
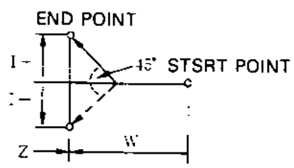
2.4.3.1 Chamfering command (G11)

G11	{ X (U).....K..... / Z (W).....I..... }	F (E).....;

This command instructs chamfering the corners of workpiece.
Only single-axis commands are available for X- or Z-axis.

Table 2.4.4 shows the meanings of command Addresses

Table 2.4.4 Meaning of Chamfering Command Addresses

X-axis Chamfering	Z-axis Chamfering
G11 X (U)...K...F (E)...;	G11 Z (W)...I...F (E)...;
	
K± .... — Chamfering quantity; — Chamfering direction designation	K± .... — Chamfering quantity (radius value); — Chamfering direction designation

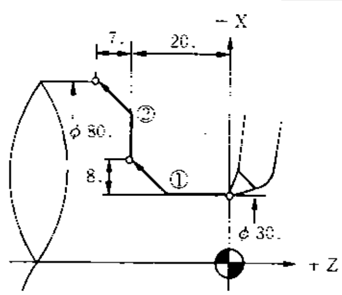
The following limitations are placed on chamfering quantities K and I:

$|K| < |U/2|$, $|I| < |W|$

Commands exceeding these limitations cause a format error.

Example of Programming

```
G00   X30.   Z 0 ;
G11   Z −20.   I 8.   F30 ;   ←①
(G11) X80.   K − 7. ;         ←②
```



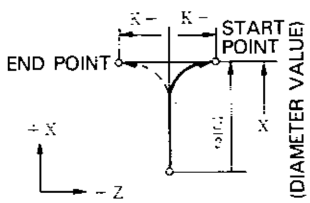
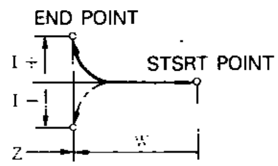
2.4.3.2 Rounding designation (G12)

$$G12 \left\{ \begin{array}{l} X\ (U).....K..... \\ Z\ (W).....I..... \end{array} \right\} \quad F\ (E).....;$$

This command instructs rounding the corners of the workpiece.
Only single-axis commands are available for X- or Z-axis.
The rounded portion is machined to a quarter circle.

Table 2.4.5 shows the meanings of command addresses

Table 2.4.5 Meaning of Rounding Command Addresses

X-axis Rounding	Z-axis Rounding
G12 X (U)...K...F (E)...;	G12 Z (W)...I...F (E)...;
	
K± .... — Rounding quantity; K± — Rounding direction designation	I ± .... — Rounding quantity; I ± — Rounding direction designation

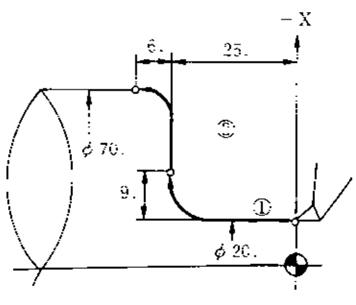
The following limitation are placed on chamfering quantities K and I:

| K | < | U/2 |, | I | < | W |

Specify I and K smaller than the total move quantity in the direction of specified axis. Commands exceeding these limitations cause a format error.

Example of Programming

```
    G00  X20.  Z0 ;
①  G12  Z-25.  I9.  F30 ;
②  (G12)  X70.  K-6.  F20 ;
```



NOTES ON G11 and G12:

1. G11 and G12 are modal G codes of 01 group. They are retained until other G codes of 01 group are specified.
2. G11 and G12 are available for single-axis. Specifying double-axis addresses in an identical block causes an alarm.

   (Example)

   G12 X....W....K....; ← Alarm "0445"
3. Blocks with I or K not specified or blocks with I or K of zero cannot be specified in G11 or G12 mode. If specified, an alarm "0445" will occur.
4. The nose radius compensation function * is available for blocks with G11 or G12 specified.
5. Compound fixed cycle: G70 to G73 finishing shape command
6. Chamfering command by G01 code

   Chamfering commands can be instructed using G01 code instead of G11.

   $$\text{G01} \begin{Bmatrix} \text{X (U)……K……} \\ \text{Z (W)…… I……} \end{Bmatrix} \quad \text{F (E)……} ;$$
7. Rounding command by G01 code

   Rounding can be instructed using G01 code instead of G12. In this case, I or K must be replaced by R.

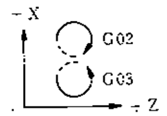
   $$\text{G01} \begin{Bmatrix} \text{X (U)……R……} \\ \text{Z (W)…… R……} \end{Bmatrix} \quad \text{F (E)……} ;$$

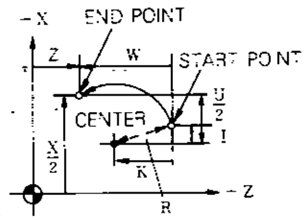
2.4.4 Circular Interpolation (G02, G03, G22, G23)

Circular interpolation is instructed with G02 and G03. Commands in **Table 2.4.6** are required for circular interpolation.

Table 2.4.6 Commands in Circular Interpolation

	Meaning
G02	Clockwise circular interpolation (CW)
G03	Counterclockwise circular interpolation (CCW)
X (U)	X-axis coordinate of end point of circular arc (diameter value)
Z (W)	Z-axis coordinate of end point of circular arc
I	X-axis distance (radius value) from start point to circular arc center
K	Z-axis distance (radius value) from start point to circular arc center
R	X-axis distance (incremental value) from start point to circular arc center
*Y (V)	Y-axis coordinate of end point of circular arc
*J	Y-axis distance from start point to circular arc center





G02 (G03) X (U)...(W)...I...K...(R...) F (E)... ;

With this command, the tool is controlled along the specified circular path on the ZX plane at a tangential speed specified by F (E).

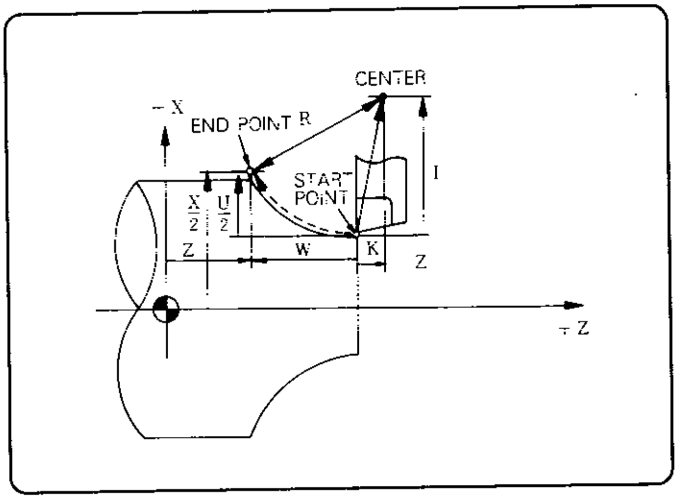


Fig. 2.4.4 Circular Interpolation

NOTE When Y-axis is provided as an additional axis, arcs on the XY or ZY plane can also be instructed by specifying the plane by G17 to G19.

(1) End point of circular arc

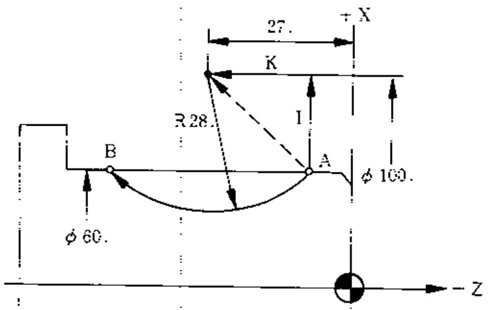
The end point of the circular arc may be specified in absolute or incremental values according to address specification or program designation.

NOTE The center of the circle is always programmed in incremental values from the start point.

(2) When a circular arc extends onto more than one quadrant, machining can be per formed by commands in a single block.

Example of Programming

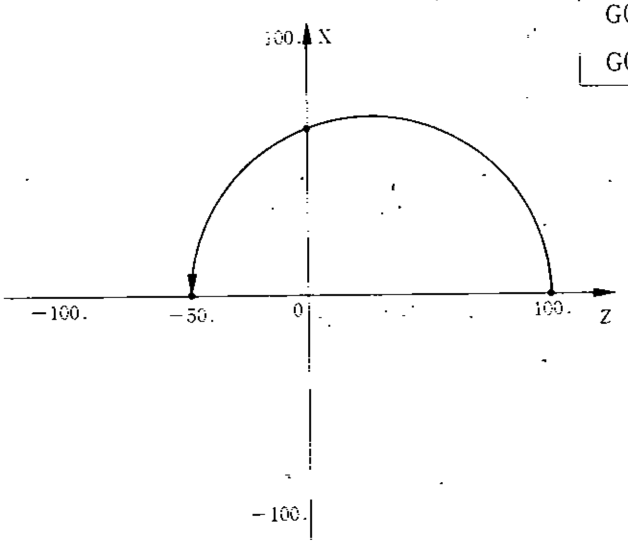
```
:
G01  Z······F······;
G02  X60.  Z-46.6  I20.  K-19.596  F······;
```



Coordinates of Center of Circular Arc	(10000, 2700)
Value of I	$\frac{100-60}{2} = 20$mm
Value of K	$-\sqrt{28^2-20^2} = -\sqrt{384}$ $= -19.596$ mm

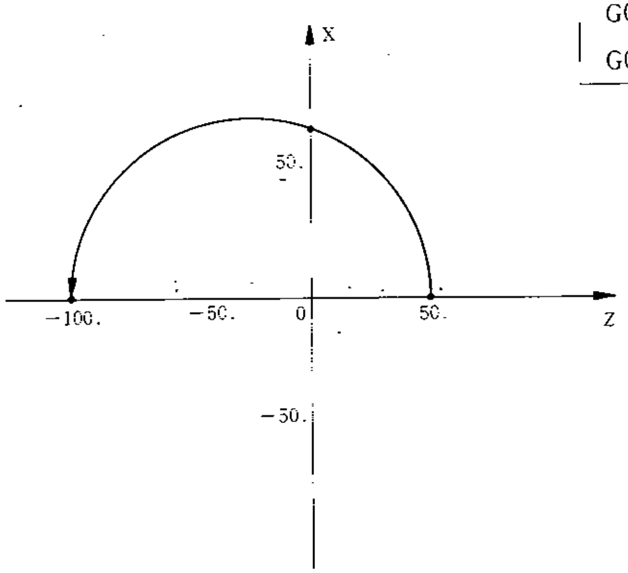
(3) When the end point of a circular arc is not on the circumference, the tool is moved as shown in Fig. 2.4.5.

```
G01  Z100.  X0   F10. ;
G03  Z-50.  K-100. ;
```



(a) Where end point is on circumference

```
G01  Z50.  X0 ;
G03  Z-100.  K-50. ;
```



(b) Where end pont is out of circumference

Fig. 2.4.5 When the End Point of A Circular Arc is not on the Circumference

(3) Circular interpolation radius (R) designation

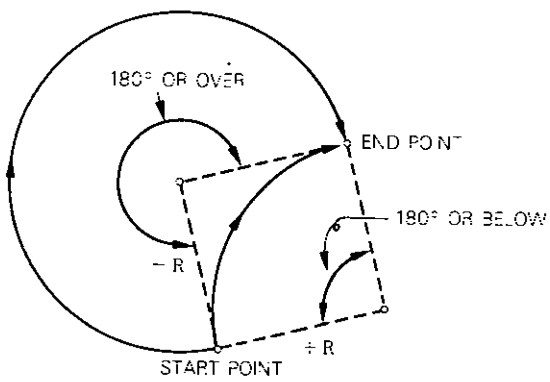
Instead of the coordinates I and K of the center of the circle, the radius can be directly specified with an R Command. This is called circular interpolation with radius R designation mode.

In this case,

- When $R > 0$, a circular arc with the center angle less than 180°, is specified, and
- When $R < 0$, a circular arc with the center angle larger than 180° is specified

Example of Programming

G02 X(U)······Z(W)······R±······F(E)······ ;



NOTE

A circular interpolation radius can also be specified using G22 or G23 instead of G02 and G03. The formats of other than G codes are the same as those of G02 and G03.

G02 corresponds to G22 and G03 to G23.

G23 and G23 cannot specify a center by I and K. If specified, an alarm "0162" will occur.

2.4.5 Polar Coordinate Interpolation (G126, G127)*

In processing with the linear axis (X-axis) and rotary axis (C-axis), assume that the C-axis is a linear axis orthogonal with the X-axis. This enables a program for machining the workpiece to any shape with the X-axis and C-axis to be created easily on the X-C orthogonal coordinate system. In this case, both absolute command (X, C) and incremental commands (U, H) can be used.

G126 ; ... Polar coordinate interpolation mode On

G127; ... Polar coordinate interpolation mode Off

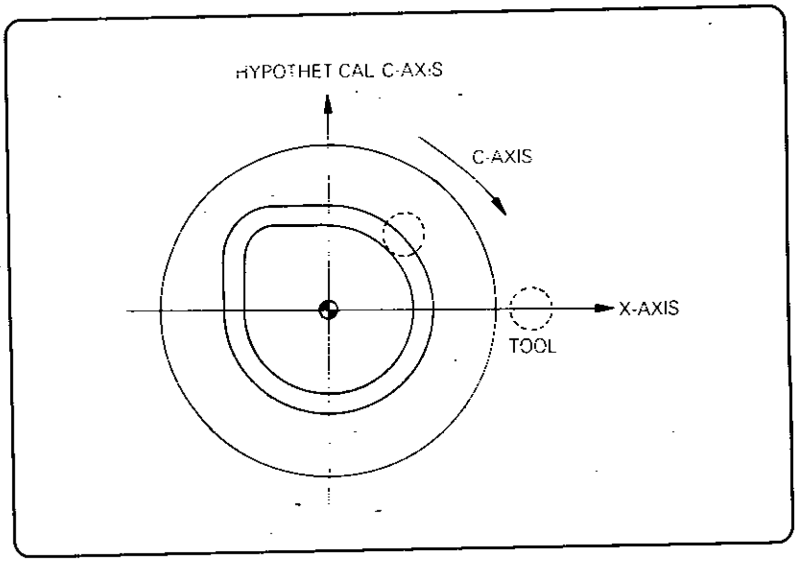


Fig. 2.4.6 Coordinate System of Polar Coordinate Interpolation

(1) G126 Command

When G126 is instructed, the polar coordinate interpolation Mode becomes in effect and hypothetical coordinate system is set with the origin of absolute coordinate system as the origin of X-Y plane. Polar coordinate interpolation is performed on that plane.

When G126 is instructed, polar coordinate interpolation is started regarding the current position of C-axis as "0."

NOTE Before G126 is instructed, the C-axis must be returned to the origin of absolute coordinate system.

(2) Tool radius compensation in G126 mode

In G126 mode, tool diameter compensation can be applied. Tool diameter compensation must be turned on or off in polar coordiante compensation mode.

Tool diameter compensation is effective only in the polar coordinate interpolation or cylindrical interpolation mode.

(For tool diameter compensation, see Par. 2.4.6.)

(3) Linear interpolation (G01) and circular arc interpolation (G02/G03)

In the polar coordiante interpolation mode, machining by linear interpolation (G01) and circular interpolation (G02/G03) can be performed.

NOTE Circular interpolation in the polar coordinate interpolation mode can be instructed only on the X-C plane. In other planes, circular interpolation commands cause an alarm.

(4) Feedrate F

Feedrate F is the speed (mm/min, inch/min) of the tool on the X-C plane.

(5) Selection of diameter/radius designation

Diameter/radius designation can be selected for X-axis and C-axis commands in the polar coordiante interpolation mode.

pm1000 D1
- 0: X-axis and C-axis commands select diameter designation.
- 1: X-axis and C-axis commands select radius designation.

(6) G codes permitted in G126 mode

G codes permitted in G126 mode are as shown below.

(G00), G01, G02, G03, G04
G10, G22, G23,
G40, G41, G42
G65, G66, G67,
(G90, G91),
G98, G134

Other commands, if specified, cause an alarm.

NOTES
1. G00 can specifiy the Z-axis only.
2. G90 and G91 are effective only in Gcode.
3. G134 is available for the M command only.

Example of Programming

```
00001 ;
G98 ;
T0101 ;
G00  X120.0  C ;  ←  Positioned at the start point of machining
G126 ;  ←  Turns on polar coordinate interpolation
G01  G42  X40.0  F100.0 ;        ┐
G03  X0  C40.0  I-20.0 ;         │
G01  X-25.0 ;                    │  Polar coordinate interpolation
G03  X-40.0  C25.0  K-15.0 ;     │  Shape processing program
G01  C0 ;                        │
G03  X20.0  I20.0 ;              │
G01  G40  X120.0 ;               │
G127 ;                           ┘
M30 ;  ←  Turns off polar coordinate interpolation
```

(7) Polar coordinate minus specifications

When the sign of X-axis is different from that of normal machines, polar coordinate minus specifications can be selected by a parameter.

The polar coordinate minus specifications state that the sign of polar coordinate X – X-axis of hypothetical C plane is reversed. Fig. 2.4.7 shows the coordinate system in which programs are created.

Whether the normal specifications or the X minus specifications are used can be selected by the parameter pm4019 D1.

Parameter pm4019 D1 = 1: X minus specifications
= 0: Normal specifications

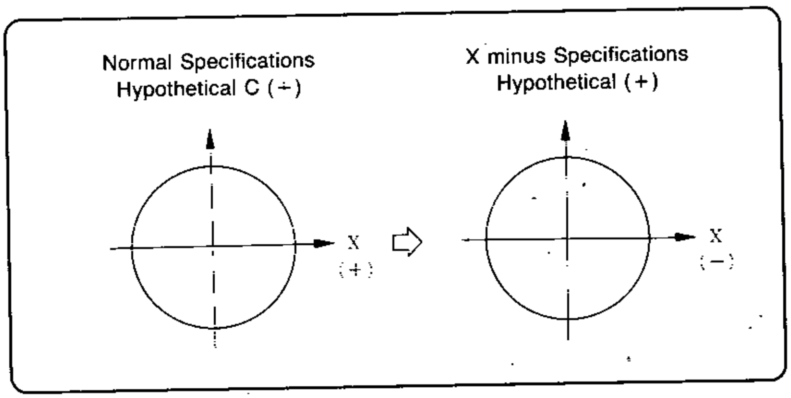
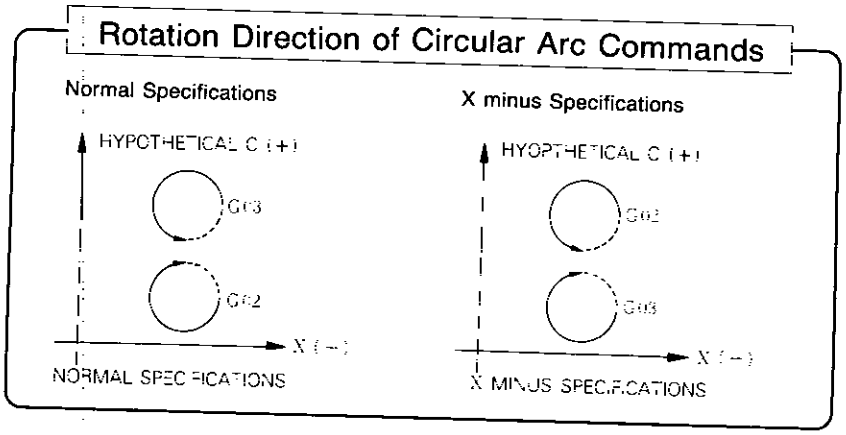
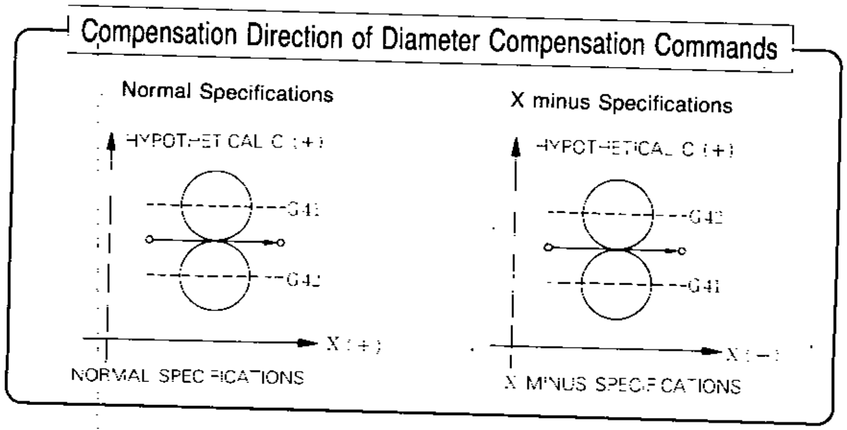


Fig. 2.4.7 Polar Coordinate Minus Specification Coordinate System

NOTE When the polar coordinate X minus specifications are selected, the rotation direction of circular arc commands (G02, G03) and the compensation direction of diameter compensation commands (G41, G42) are reversed for the normal specifications. Therefore, programming must be performed with that fact in mind.





Turn on the polar coordinate interpolation mode when X coordinate is plus in the case of normal polar coordinate specifications, and minus in the case of X minus specifications.

NOTES ON G126 and G127:

1. Instruct G126 and G127 in a single block.
2. G126 and G127 are modal G codes of 19 group. When G126 is instructed, the polar coordinate interpolation mode remains on until the next G127 is instructed.
3. The equipment enters G127 (polar coordinate interpolation mode off) state at power on and at reset.
4. Be sure to cancel nose radius compensation before instructing G126.
5. Prohibition of mirror image

   G126 cannot be instructed when the mirror image mode is on.
   Mirror image cannot be applied in G126 mode; an alarm would occur.
6. G98 (feed per minute) must be instructed in the polar coordinate interpolation mode.
7. G00 (G code including rapid feed cycle) cannot be instructed in the polar coordinate interpolation mode. For positioning, cancel the polar coordinate interpolation mode.
   G00 can be instructed outside the X-C plane.
8. In the polar coordinate interpolation mode, interpolation is performed on the X-C hypothetical plane.
   Therefore, a feed speed on the X-C plane must be instructed.
9. Circular commands in the polar coordinate interpolation mode specify X-axis by I and C-axis by K to represent the address of circle center.
10. T, S, and M commands

    T and S cannot be instructed in the polar coordiante interpolation mode. M can be instructed.
11. The spindle function is disabled in the polar coordiante interpolation mode.
12. In the polar coordinate interpolation mode, manual absolute is fixed to OFF.
13. Programs cannot be restarted in the polar coordinate interpolation mode.

    However, programs can be restarted in a range containing the polar coordinate interpolation mode.
14. Since C-axis speed becomes infinite when a command to pass the center of the coordinate is specified in the polar coordinate interpolation mode, Alarm 483 "command error in the polar coordinate interpolation mode".

2.4.6 Cylindrical Interpolation (G124, G125)*

The cylindrical interplation function enables processing on a cylindrical workpiece to be programmed in plane image using a cylindrically expanded coordinate system in which the side of the cylinder is expanded.

This function can be instructed by either of absolute commands (C, Z) and incremental commands (H, W).

Cylindrical interpolation is instructed using the following G code:

G124 C....... ;	← Turns on the cylindrical interpolation mode. C is the radius of workpiece to be processed.
G125	← Turns off the cylindrical interpolation mode.

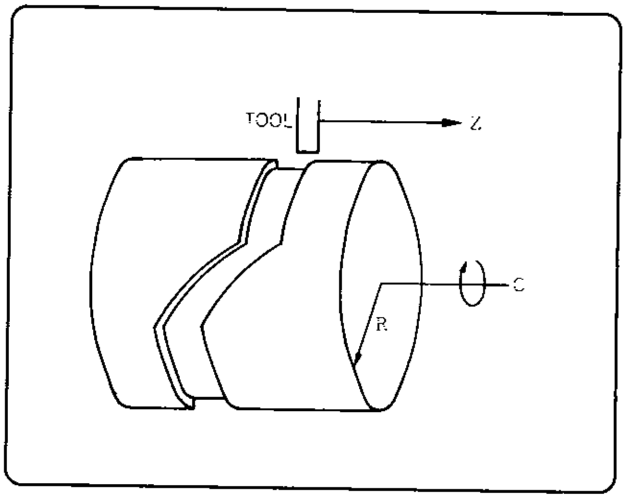


Fig. 2.4.8 Coordinate System of Cylindrical Interpolation

(1) Tool diameter compensation in cylindrical interpolation mode

In the cylindrical interpolation mode, tool diameter compensation can be applied. Tool diameter compensation must be turned on or off in cylindrical interpolation mode.

Tool diameter compensation is effective only in the polar coordinate interpolation or cylindrical interpolation mode.

(2) Machining by linear interpolation (G01) and circular arc interpolation (G02/G 03)

In the cylindrical interpolation mode, machining by linear interpolation (G01) and circular interpolation (G02/G03) can be performed.

(3) Circular interpolation in the cylindrical interpolation mode

Circular interpolation in the cylindrical interpolation mode can be instructed only on the C-Z plane. In other planes, circular interpolation commands cause an alarm.

(4) Feedrate F

Feedrate F is the speed (mm/min, inch/min) of the tool on the C-Z plane.

(5) G codes permitted in G124 mode

G codes permitted in G124 mode are as shown below.

(G00), G01, G02, G03, G04
G10, G22, G23,
G40, G41, G42,
G65, G66, G67,
(G90, G91),
G98, G134

Other commands, if specified, cause an alarm.

NOTES
1. G00 can specify the X-axis only.
2. G90 and G91 are effective only in G code.
3. G134 is available for the M command only.

Example of Programming

```
0100 ;
G98 ;
T010 ;
G00  X44.0  C0 ;        ← Positioned at the start point of machining
G124  C45.0 ;           ← Turns on cylindrical interpolation
G01  G42  Z47.5  F100 ; ┐
C60.0 ;                 │
Z32.5  C120.0 ;         │
C240.0 ;                │ Cylindrical interpolation
G03  Z40.0  C249.549  R7.5 ; │ Shape processing progarm
G02  Z47.5  C259.099  R7.5 ; │
G01  C360.0 ;           │
G40  Z44.0 ;            ┘
G125 ;                  ← Turns off cylindrical interpolation
M30 ;
```

NOTES ON G124 and G125:

1. Instruct G124 and G125 in a single block.
2. G124 and G125 are modal G codes of 19 group. When G124 is instructed, the cylindrical interpolation mode remains on until the next G125 is instructed.
3. The equipment enters G125 (cylindrical interpolation mode off) state at power on and at reset.
4. Be sure to cancel nose radius compensation before instructing G126.
5. Prohibition of mirror image

   G124 cannot be instructed when the mirror image mode is on.
   Mirror image cannot be applied in G124 mode; an alarm would occur.
6. Circular commands in the cylindrical interpolation mode cannot specify address radius by I and K. Specify a radius value directly by address R.
7. G98 (feed per minute) must be instructed in the cylindrical interpolation mode.
8. G00 (G code including rapid feed cycle) cannot be instructed in the cylindrical interpolation mode. For positioning, cancel the cylindrical interpolation mode.
   G00 can be instructed outside the C-Z plane.
9. In the cylindrical interpolation mode, interpolation is performed on the C-Z hypothetical plane.
   Therefore, a feed speed on the C-Z plane must be instructed.
10. T, S, and M commands

    T and S cannot be instructed in the cylindrical interpolation mode. M can be instructed.
11. The spindle function is disabled in the cylindrical interpolation mode.
12. In the cylindrical interpolation mode, manual absolute is fixed to OFF.
13. Programs cannot be restarted in the cylindrical interpolation mode.
    However, programs can be restarted in a range containing the cylindrical interpolation mode.

2.4.7 Tool Diameter Compensation Functions Available in Polar Coordinate Interpolation and Cylindrical Interpolation

(1) Automatic offset of tool path

By using tool diameter compensation, tool path can be automatically offset by the specified radius of the tool (rotary tool) to be used.

(2) Specifying tool radius

A tool radius is stored in tool offset memory R and is given by specifying the offset number by T code. Set control point to zero.

(3) Specifying tool diameter compensation by G code

The following G codes are used for tool diameter compensation:

G41 (left compensation) G42 (right compensation)	Turns on tool diameter compensation mode on.
G40	Turns off tool diameter compensation mode off.

(4) Compensation plane

Compensation planes are:
X-C plane: In polar coordinate interpolation mode.
C-Z plane: In cylindrical interpolation mode

(5) Entry to tool diameter compensation mode

When G41 or G42 is instructed, the compensation mode becomes effective.

The direction of offset is on the normal of the start point of a block next to G41 or G42, and compensation is applied to the left or right in the advance direction by an offset quantity by G41 or G42.

(6) Behavior in tool diameter compensation mode

Upon entry to the compensation mode by G41 or G42, the tool moves along the offset path until G40 is instructed.

Since the path is automatically computed by the NC equipment, the part program has only to specify the shape of the workpiece.

The tool path is controlled as follows by the angle between angle blocks:

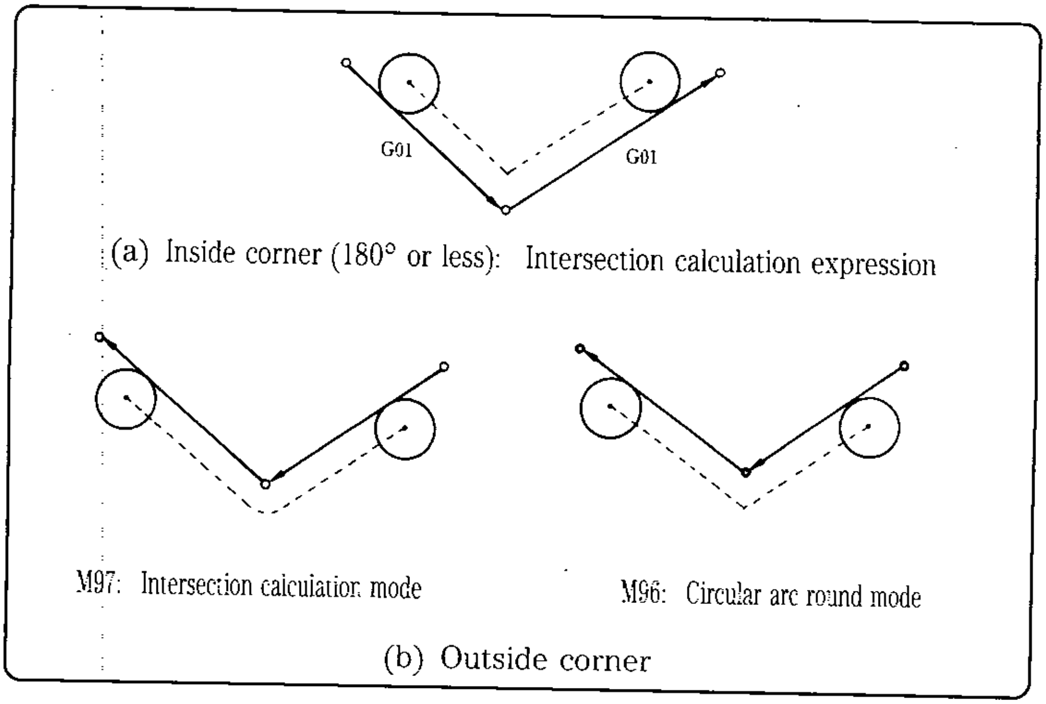


Fig. 2.4.9 Tool Path

(7) Canceling tool diameter compensation

The compensation mode is canceled by G40.

When G40 is instructed, the tool is positioned at the start point of the last block of compensation mode.

(8) Compensation modes enabling the tool diameter compensation functions

The tool diameter compensation functions are available only in the cylindrical interpolation mode or polar coordinate interpolation mode.

(9) Turning on or off the tool diameter compensation mode

Turn on or off the tool diameter compensation mode in the cylindrical interpolation mode or polar coordinate interpolation mode.

(10) Changing tool diameter

A tool diameter cannot be changed when the tool diameter compensation mode is on.

2.5 FEED FUNCTIONS

2.5.1 Rapid Traverse Rate

The rapid traverse motion is used for the motion for positioning (G00) and for the motion for manual rapid traverse (RAPID). The tool moves at a rapid traverse rate defined independently for each axis. The rapid traverse rate is determined by your machine manufacturer and preset for each axis.

Motion paths are normally not straight because the end points of each axis are reached at different times by independent axis motion.

For override of rapid traverse rates, F_0, 25%, 50%, and 100% of the basic rapid traverse rates, are available. F_0 is a constant feedrate set by a parameter (pm2447). Six levels of rapid traverse override can be defined by parameters: F_0, F_1, F_2, 25%, 50%, and 100%. F_1 and F_2 are set in parameters (pm2448, pm2449).

For F_0, either setting unit mm/min or % can be selected according to pm 2000 D3 setting.

pm2000 D3 = 0 = F_0 (pm2447) setting unit: mm/min
pm2000 D3 = 1 = F_0 (pm2447) setting unit: %

Range of setting rapid traverse rates

(1) For each axis, rapid traverse rates can be set at some suitable multiple of 0.001 mm/min (deg/min).

(2) The maximum programmable rapid traverse rate is 240,000 mm/min.

NOTE Machine tools have their own optimum rapid traverse rates. Refer to the manual provided by the machine tool builder.

2.5.2 Cutting Feed

With digits following address characters F and E, tool feedrates per minute (mm/min) are programmed at linear interpolation (G01) or circular interpolation (G02, G03).

Two types of feedrate specifications can be selected using the G codes for specifying the feed function shown in **Table 2.5.1**. Specify the G codes before instructing F and E codes.

Table 2.5.1 G Codes for Specifying Feed Function

G code	Meaning
G99	Feed per rotation (mm/rev)
G98	Feed per minute (mm/min)

For details of the G code functions, see Par. 2.5.3.

(1) F and E codes are modal, and once they are instructed, they are effective until the next time an F or E code is instructed. When switching between G98 and G99 is performed, an F or E code must be instructed newly. If not instructed, alarm "0370" will occur.

(2) In feed-per-minute (mm/min) specification, E code cannot be instructed. If in structed, alarm "0371" will occur.

2.5.2.1 Feed per rotation (G99 mode)

(1) With a numeric value following an address F or E, tool feedrate (mm/rev or inch/rev) per spindle rotation can be specified.

Table 2.5.2 shows the range of F and E code specification.

Table 2.5.2 Range of Feed-per-Rotation F and E Code Specification

		Format	Feed-per-rotation specification range
Metric Input	Metric input	F33	F0.001 to F500.000 mm/rev
		E34	E0.0001 to E500.0000 mm/rev
	Inch input	F24	F0.0001 to F19.6850 in/rev
		E26	E0.000001`E19.685000 in/rev
Inch Input	Metric input	F33	F0.001 to F1270.000 mm/rev
		E34	E0.0001 to E1270.0000 mm/rev
	Inch input	F24	F0.0001 to F50.0000 in/rev
		E26	E0.000001 to E50.000000 in/rev

Table 2.5.3 shows limits by spindle rotation speed.

Table 2.5.3 F (E) Specification Limits by Spindle Rotation Rate

	Limit value
Metric output	F (E) $\times$ S $\leqq$ 240,000 mm/min
Inch output	F (E) $\times$ S $\leqq$ 24,000 in/min

NOTES
1. X-axis speed limit values are one-half of the values shown in the above table.
2. The limit values per minute are further limited by the conditions of machines. Follow the machine tool manufacturer's manual.

(2) F command for linear and circular interpolation specifies feedrates in the direction tangential to the motion path.

```
G99  S1000 (r/min) ;
G01  U60.  W40.  F0.5 ;
```

$F \times S = 0.5\text{mm/rev} \times 1000\text{r/min}$

$= 500\text{mm/min}$

$= \sqrt{\underbrace{300^2}_{\text{X-AXIS SPEED COMPONENT}} + \underbrace{400^2}_{\text{Z-AXIS SPEED COMPONENT}}}$

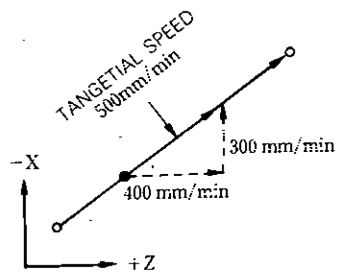


Fig. 2.5.1 F Command in Simultaneous Two-Axis Linear Interpolation (Feed per Rotation)

```
G99  S1000 (r/min) ;
G03  U.....W.....I.....F0.2
```

$F \times S = 0.2 \times 1000 = 200\text{mm/min}$

(mm/rev) (r/min)

$= \sqrt{fx^2 + fz^2}$

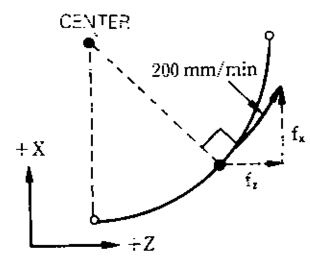


Fig. 2.5.2 F Command in Simultaneous Two-Axis Circular Interpolation (Feed per Rotation)

NOTES

1. If F0 is programmed, it is regarded as an input error.
2. Do not program F commands with minus numerals; alarm "0102" will occur.
3. Feedrate in the X-axis direction are rates obtained by radius values.

2.5.2.2 Feed per minute (G98 mode)

(1) With a numeric value following an address F, tool feedrate (mm/minute or inch /mm) per minute can be specified.

F60 format or F63 format (at metric input) can be selected by parameter pm2004 D0.

Table 2.5.4 shows the range of F code specification.

Table 2.5.4 Range of Feed-per-Minute F Code Specification

			Format	Linear Axis Feed-per-minute Specification Range	Rotary Axis Feed-per-minute Specification Range
pm2004 D0=0	Metric output	Metric input	F60	F1 to F240000 mm/min	F1 to F240000 deg/min
		Inch input	F52	F0.01 to F94488.18 in/min	F0.01 to F240000.00 deg/min
	Inch output	Metric input	F60	F1 to F609600 mm/min	F1 to F240000 deg/min
		Inch input	F52	F0.01 to F24000.00 in/min	F0.01 to F240000.00 deg/min
pm2004 D0=1	Metric output	Metric input	F63	F0.001 to F240000.00 mm/min	F0.001 to F240000.000 deg/min
		Inch input	F54	F0.0001 to F94488.1890 in/min	F0.0001 to F240000.0000 deg/min
	Inch output	Metric input	F63	F0.001 to F609600.000 mm/min	F0.001 to F240000.000 deg/min
		Inch input	F54	F0.0001 to F24000.0000 in/min	F0.0001 to F24000.0000 deg/min

NOTES
1. X-axis speed limit values are one-half of the values shown in the above table.
2. The limit values per minute are further limited by the conditions of machines. Refer to the machine tool manufacturer's manual.

(2) F command for linear and circular interpolation at simultaneous two-axis control specifies feedrates in the direction tangential to the motion path.

```
G98 ;
G01 U60.W40.F500. ;
.........
```

$F=500=\sqrt{300^2+400^2}$

(mm/min)

Z-AXIS SPEED COMPONENT

X-AXIS SPEED COMPONENT

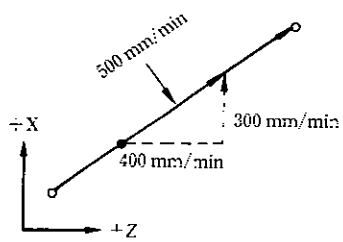


Fig. 2.5.3 F Command in Simultaneous Two-Axis Linear Interpolation (Feed per Minute)

G98 :
G03 X.....Z.....I.....F200.;
.....

$F=200=\sqrt{fx^2+fz^2}$
(mm/min)

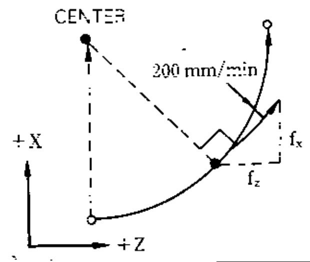


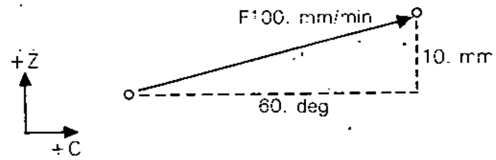
Fig. 2.5.4 F Command in Simultaneous Two-Axis Circular Interpolation (Feed per Minute)

NOTES

1. If F0 is programmed, it is regarded as an input error.
2. Do not program F commands with minus numerals; alarm "0102" will occur.
3. Feedrates in the X-axis direction are rates obtained by radius values.

(3) F command for interpolation at the rotary and linear axes control specified feedrates in the diraction tangential to the motion path.

G98;
G01 W10. H60. F100.;
.....



a) Metric input (F6.0)

Distance $= \sqrt{1000^2+60000^2} = 60827,625$

Z-AXIS SPEED COMPONENT

C-AXIS SPEED COMPONENT

$$\text{Time} = \frac{60827.625}{100000} = 0.6082\ (\text{min}) = 36.5\ (\text{s})$$

b) Inch input (F5.2)

Distance $= \sqrt{100000^2+60000^2} = 1166190,0379$

Z-AXIS SPEED COMPONENT

C-AXIS SPEED COMPONENT

$$\text{Time} = \frac{1166190.0379}{1000000} = 0.1166\ (\text{min}) = 6.9\ (\text{s})$$

(4) Individual command at the rotary axis control specifies feedrates by units of input system (metric/inches).

For inputting inches, the feedrates are specified by the parameter as.

		Format	pm2004 D7=0	pm2004 D7=1
pm2004 D0=0	Metric input	F60	1=1 deg/min	1=1 deg/min
	Inch input	F52	1=0.1 deg/min	1=0.01 deg/min
pm2004 D0=1	Metric input	F63	1=0.001 deg/min	1=0.001 deg/min
	Inch input	F54	1=0.001 deg/min	1=0.0001 deg/min

pm2004 D0

Feed command

0: Metric input → F60 (mm/min)
Inch input → F52 (in/min)

1: Metric input → F63 (mm/min)
Inch input → F54 (in/min)

pm2007 D7

Feedrates for rotary axis individual command

0: F52 → 1=0.1 deg/min
F54 → 1=0.001 deg/min

1: F52 → 1=0.01 deg/min
F54 → 1=0.0001 deg/min

2.5.3 Switching between Feed per Minute and Feed per Rotation (G98/G99)

These G codes specify feeding per minute or feeding per rotation before the F (E) code for feeding.

(1) G98 ;

This command executes subsequently issued F codes as feeding per minute.

Table 2.5.5 Meaning of G98 Code

G98	Meaning
Metric input	mm/min
Inch input	in/min

(2) G99 ;

This command executes subsequently issued F (E) codes as feeding per rotation.

Table 2.5.6 Meaning of G99 Code

G99	Meaning
Metric input	mm/rev
Inch input	in/rev

(3) **G98 and G99 are modal G codes.**

G98 and G99 are modal G codes and effective until specification is changed.

(4) **Specification at switching between G98 and G99**

Switching between G98 and G99 cancels F (E) code specified previously. Therefore, whenever the switching has been made, be sure to specify F (E) code.

(5) **Switching of initial state at power on**

The initial state at power on can be set to G98 or G99 by parameters.

Table 2.5.7 Parameter pm4000 and Initial State

Parameter	Initial G code
pm4000 D2=0	G98
pm4000 D2=1	G99

2.5.4 Screw Cutting and Continuous Screw Cutting (G32)

G32 X (U).......Z (W)......F(E)....... ;

With this command, a straight screw, taper screw, and scroll screw can be cut by a lead specified in an F or E code all the way to the absolute coordinate (X, Z) point or incremental position (U, W) point.

(1) Range covered by F and E codes

Table 2.5.8 shows the range of F and E codes specifying screw lead.

Table 2.5.8 Range of F and E Codes for Screw Cutting

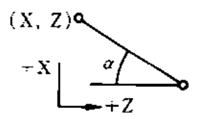
		Format	Range of F and E Specification for Screw Cutting
Metric Output	Metric input	F33	F0.001 to F500.000 mm
		E34	E0.0001 to E500.0000 mm
	Inch output	F24	F0.0001 to F19.6850 in
		E26	E0.000004 to F19.685000 in
Inch Input	Metric input	F33	F0.001 to F1270.000 mm
		E34	E0.0003 to E1270.0000 mm
	Inch input	F24	F0.001 to F50.0000 in
		E26	E0.000010 to E50.000000 in

NOTE The F code is used for normal screw cutting.
The E code is used for precise screw cutting.

(2) Direction of screw lead

Table 2.5.9 shows the direction of lead specified by F and E codes.

Table 2.5.9 Lead Direction

Condition of taper angle		Direction of screw lead
	a ≦ 45°	Specifies lead in Z-axis direction.
	a > 45°	Specifies lead in X- axis direction.

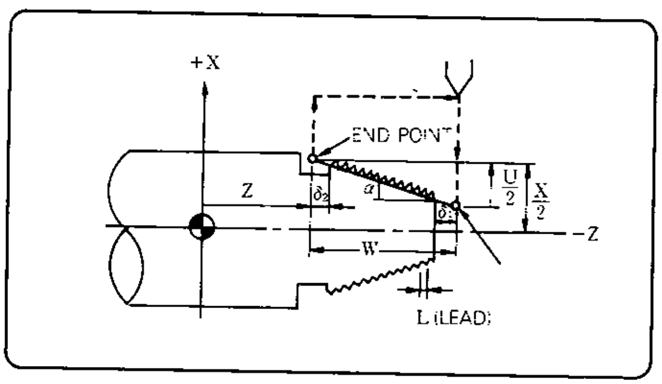


Fig. 2.5.5 Screw Cutting

(3) Limit values by spindle rotation speed S

F and E code specifications are limited by spindle rotation speed S as shown in **Table 2.5.10**.

Table 2.5.10 Limit Values by Spindle Rotation Speed S

	Limit value
Metric output	F (E) × S ≦ 240,000 mm/min
Inch output	F (E) × S ≦ 24,000 in/minin

The upper limit of X-axis speed component is one-half of the above.

(4) Screw specification

Table 2.5.11 shows the methods for specifying individual screws.

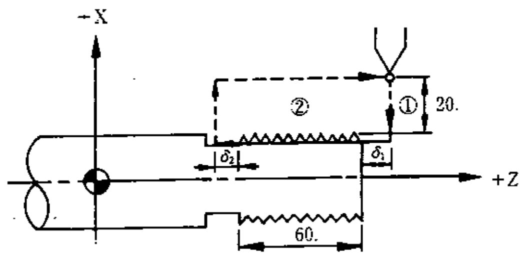
Table 2.5.11 Screw Specification

Type of screw		Specification
Straight screw	Normal	G32 Z (W)...F... ;
	Precise	G32 Z (W)...E... ;
Taper screw	Normal	G32 X (U)...Z (W)...F... ;
	Precise	G32 X (U)...Z (W)...E... ;
Scroll screw	Normal	G32 X (U)...F... ;
	Precise	G32 X (U)...E... ;

• Straight screw

Screw lead L = 5.0mm
δ_1 = 5.0mm
δ_2 = 3.0mm
Cutting quantity = 1.0mm

```
G00  U42. ;        ← ①
G32 W−68. F5.0 ;   ← ②
G00 U  42. ;
    W  68. ;
    U−44. ;
G32 W−68. ;
G00 U  44. ;
    ⋮
```

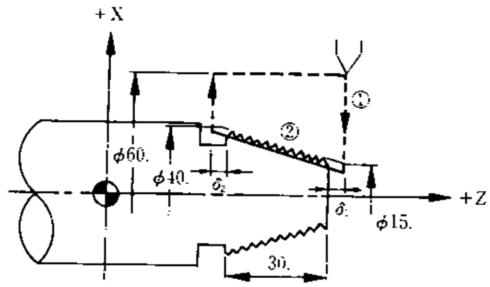


Example of Programming

• **Taper screw**

Screw lead L=4.0mm
δ_1=3.0mm
δ_2=2.0mm
Cutting quantity=1.0mm

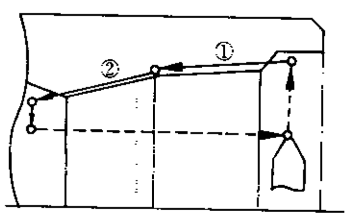
```
G00  X13.
G32  X38. W-35. F4.0 ;
G00  X60. ;
     W35. ;
     X11. ;
G32  X36. W-35. ;
G00  X60. ;
  ⋮
```



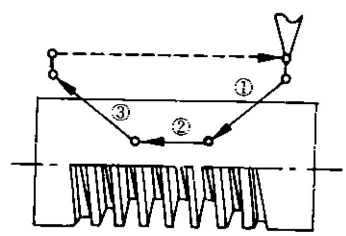
(5) Continuous screw cutting

Since this equipment has fast buffer registers, continuous screw cutting can be specified as shown in **Fig. 2.5.6**.

```
G33    X(U)...Z(W)...F(E)... ;  ← ①
(G32)  X(U)...Z(W)... ;         ← ②
(G32)  X(U)...Z(W)... ;         ← ③
  ⋮
```



(a) Improved pipe coupling



(b) Worm screw

Fig. 2.5.6 Continuous Screw Cutting

Since a stop time between screw cutting blocks is zero, continuous screws can be cut smoothly.

NOTE Do not change screw lead specification F or E midway: if this were done, bad screw cutting would occur in the neighborhood of block joints.

(6) Slack in imperfect screw portion (δ_1, δ_2)

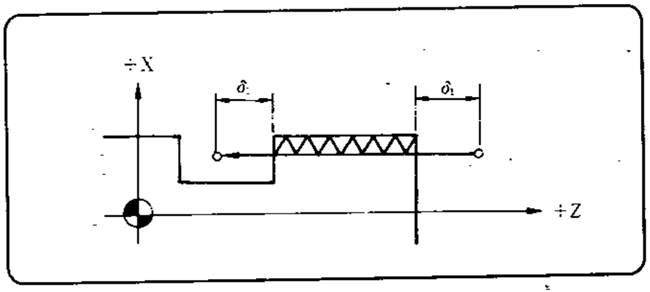


Fig. 2.5.7 Slack in Imperfect Screw Portion

The slacks δ_1 and δ_1 are obtained by the approximate expression in Table 2.5.12.

Table 2.5.12 Slacks in Imperfect Screw Portion

	Approximate expression	Meaning
δ_1	$\delta_1 > \frac{L \bullet S}{60 \bullet K}\ (\ln \frac{1}{a} - 1)$	L (mm): Screw lead S (rpm): Spindle rotation speed K: Constant. Usually 30 a (−): Screw precision
δ_2	$\delta_2 > \frac{L \bullet S}{60 \bullet K}$	$= \frac{\Delta L}{L}$ .... (Lead error) In: Natural logarithm (log ϵ)

a	1/50	1/100	1/150	1/200	1/250	1/300
$(\ln \frac{1}{a} - 1)$	2.91	3.61	4.01	4.29	4.52	4.70

(Example of computation)

Screw lead L = 3.0 mm
Spindle speed S: 50 r/min
Screw precision = 1/100

δ_1 and δ_2 are obtained as follows when the above conditions are given:

$$\delta_1 > \frac{L \bullet S}{60 \bullet K} (\ln \frac{1}{a} - 1)$$

$$= \frac{3.0 \times 500}{60 \bullet K} \times 3.61 = 3.0\text{mm}$$

$$\delta_2 > \frac{L \bullet K}{60 \bullet K} = \frac{3.0 \times 500}{60 \bullet K} = 0.83\text{mm}$$

NOTES ON G32:

1. **How to take up slack in imperfect screw portion**

   Since a lead error occurs at the start and end of screw cutting, slack shown by δ_1 and δ_2 in **Fig. 2.5.8** must be taken. See (6) for how to take up the slack in imperfect screw portions.

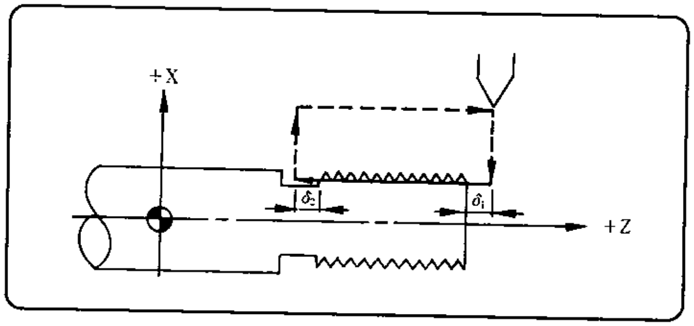


Fig. 2.5.8 Start and End of Screw Cutting

2. **Spindle rotation speed**

   The spindle rotation speed must be constant until the cutting of one screw is terminated. Otherwise, the precision may not be maintained due to a servo delay.

3. **Screw rounding-off in G32 specification**

   G32 specification does not permit screw rounding-off. If rounding-off is required, use G92 (✱ or G76).

4. **Operation disregarded during screw cutting**

   The following operations including G32 are disregarded during screw cutting:

   a. Feed override operation ... Regarded as 100%
   b. Temporary stop (FEED HOLD) operation

5. **Screw cutting G32 in G98 mode**

   Screw cutting G32 must not be instructed in G98 (feed per minute) mode. If insturcted, Alarm "0452" will occur.

6. **Screw cutting command during dry run**

   Screw cutting commands during dry run specify movement at a manual continuous speed.

7. **M code command**

   Do not issue continuous screw cutting commands with M code specified. If issued, the program would wait for a start point in that block and continuous screws could not be cut.

2.5.5 Multiple Thread Screw Cutting (G32)*

Multiple thread screws (having more than one thread in a lead) can be cut without shifting a cutting start position.

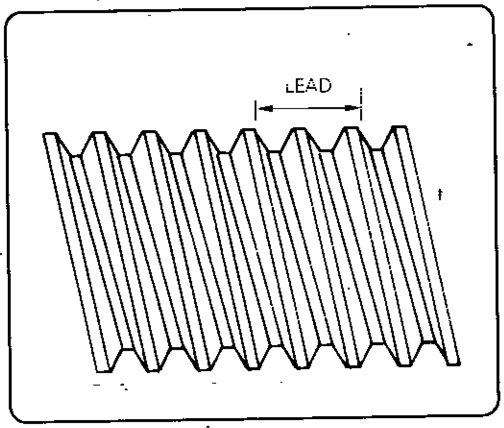


Fig. 2.5.9 Double-Threaded Screw

Since screw cutting operation starts feed synchronously with a start point pulse (one pulse per rotation) from the spindle pulse generator attached to the spindle, a cutting start point on the workpiece circumference is always the same and screws can thus be cut.

A multiple thread screw is cut in such a way that screw cutting is performed by feeding synchronized with a start point pulse from the spindle pulse generator, then feeding is started after the spindle revolves a given angle after the start point pulse is emitted.

G32 X(U)……Z(W)……F(E)……B… ;

With this command, the spindle revolves the angle specified in B after a start point pulse from the spindle pulse generator, then screw cutting is started from X (U) to Z (W) point with a lead specified by the F or E command.

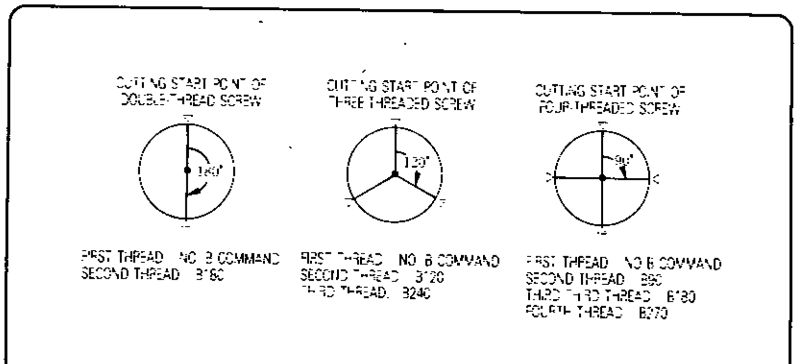
(1) Address B specified in multiple thread screw cutting

Least input increment 0.001°	
Command range	$0 \leqq B < 360.00$

When decimal point input is used, B1 = 1° is set.
The B command is non-modal and is effective only in the block in which it is issued.

(2) Number of threads of multiple thread screw and B command

A cutting start point on the workpiece circumference is normally on one of the points in which the circumference (360°) is divided equally by the number of threads.



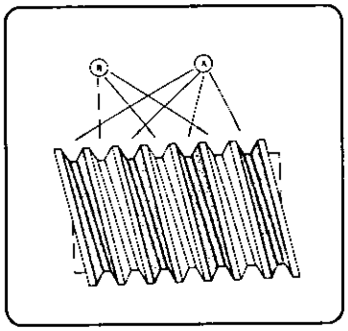
(a) Double-threaded Screw (b) Three-threaded Screw (c) Four-threaded Screw

Fig. 2.5.10 Number of Threads of Multiple-Thread Screw and B Command

Example of Programming

- Double-threaded screw cutting

```
G00  U....... ;      ┐
G32  W....... ;      │
G00  U....... ;      │ Screw cutting
W....... ;           │ of (A)
U....... ;           │
G32  W....... ;      ┘
  :
G00  U....... ;
G32  W.......B180. ; ┐
G00  U....... ;      │ Screw cutting
W....... ;           │ of (B)
U....... ;           ┘
G32  W.......B180. ;
  :
```



NOTES ON G32:

1. **Rotation detecting pulse for B command**

   Since a spindle rotation angle from a start point pulse by B command uses rotation detecting pulses (4096 pulses per rotation), the minimum detection unit is about 0.0879° (360°/4096 pulses) per pulse.

   For the B command, the number of rotation detecting pulses may have an error of ±1 pulse.

2. **Specifying a spindle rotation angle from start point pulse**

   A spindle rotation (whether forward or backward) angle from a start point pulse is specified as an angle from 0° to 360° by a B command.

3. **Note on B command for multiple-thread screw cutting**

   If the B command for multiple-thread screw specifies out of the legal range (0° to 360.000°), an error "0453" will occur.

4. **Note on B command for multiple-thread screw cutting**

   When the B command for multiple-thread screw cutting is issued, continuous screws cannot be cut.

   G32W.......90

   G32W....... ← Since the program waits for a start point in this block, continuous screws cannot be cut.

2.5.6 Variable Lead Screw Cutting (G34)*

| G34 X(U) Z(W).......K.......F(E)... ; |

With this command, a variable lead screw specified by address K can be cut by the quantity of lead per rotation of the screw.

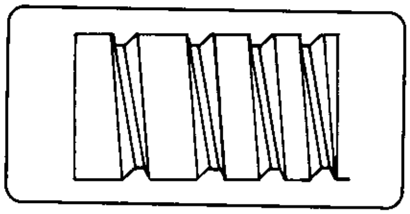


Fig. 2.5.11 Variable Lead Screw

(1) Range of K specification in variable lead screw cutting

Least input increment 0.0001 mm/rev or 0.00001 inch/rev
(Metric input) (Inch input)

Specification range

a. The speed at an end point must not exceed the specification range (500 mm/rev for metric output and 50 inch/rev for inch output) of feedrate.

b. The accumulation of movements due to changes in lead must not exceed the values shown in **Table 2.15.3.**

Table 2.5.13 Upper Limit on Accumulation of Movements due to Changes in Lead

	Upper limit
Metric output	4194.303 mm
Inch output	419.4303 in

c. Lead change speeds must not exceed the values shown in **Table 2.15.4.**

Table 2.5.14 Upper Limit on Lead Change Speed

	Upper limit
Metric output	5,400 mm/min
Inch output	540 in/min

d. A lead must not be minus.

(2) Additional explanation

Arithmetic expressions for checking the variable lead screw cutting (G34) command

Limitation on K command

a. The feedrate at an end point must not exceed the values shown in **Table 2.5.15**.

Table 2.5.15 Upper limit on Feedrate at End Point

	Upper limit
Metric output	500 mm/rev
Inch output	50 in/rev

b. The feedrate at an end point must not be minus.

c. The accumulation of movements due to changes in lead must not exceed the values shown in **Table 2.15.16**.

Table 2.5.16 Upper Limit on Accumulation of Movements due to Changes in Lead

	Upper limit
Metric output	4194.303 mm
Inch output	419.4303 in

d. Lead change speeds must not exceed the values shown in **Table 2.15.17**.

Table 2.5.17 Upper Limit on Lead Change Speed

	Upper limit
Metric output	5400 mm/min
Inch output	540 in/min

Checking limits by arithmetic expressions

F : Fixed lead command (mm/rev or inch/rev)
K : Variable lead command (mm/rev or inch/rev)
W : Z-axis distance (mm or inch) from a start point to an end point (mm or inch) (X-axis distance U in the case of end screw)
S : Spindle rotation speed (r/min)
N : Spindle rotation speed (rev) required from a start point to an end point

$$N = \frac{-(F+K/2) + \sqrt{(F+\frac{K}{2})^2 + 2 \bullet K \bullet W}}{K}$$

Limit of a

$$F + \frac{K}{2} + KN \leqq 500.000 \text{ mm/rev or } 50.000 \text{ inch/rev}$$

Limit of b

$$(F + \frac{K}{2})^2 + 2KW > 0$$

Limit of c

$$\frac{1}{2} KN^2 \leqq 4194.303 \text{ mm or } 419.4303 \text{ inches}$$

Limit of d

$$S/60 \bullet K \bullet N \leqq 5{,}400 \text{ mm/min or } 540 \text{ inch/min}$$

NOTES

1. **Continuous block screw cutting**

   In continuous block screw cutting by the variable lead screw cutting command, command pulses break at block joints.

2. **K command**

   If the K command exceeds the legal range, an error "0450" occurs.

3. **When the G34 command is issued during dry run, if the parameter (pm2000 D1) is set to 1, the tool only moves at a specified manual continuous feedrate.**

4. **When the parameter (pm1000 D0) is set to 1, the least input increment of K command is 0.001 mm/rev or 0.0001 inch/rev.**

5. **Address B command in G34 block**

   Specifying address b in a G34 block causes an error "0450."

2.5.7 Automatic Acceleration and Deceleration

Accel/decel for rapid traverse and cutting feed are automatically performed.

2.5.7.1 Accel/decel of rapid traverse and manual feed

(1) In the following operation, the pattern of automatic accel/decel is linear.

- Positioning (G00)
- Manual rapid traverse (RAPID)
- Manual continuous feeding (JOG)
- Manual handle feeding (HANDLE)

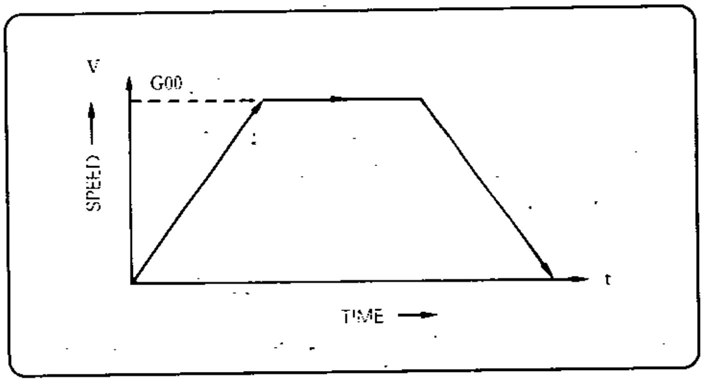


Fig. 2.5.12 Linear Accel/decel Speed

NOTE Rapid traverse rate and accel/decel constant of rapid traverse rate can be set by parameter. (pm2461 to pm2468, pm2801 to pm2808)

2.5.7.2 Accel/decel of cutting feed

(1) Automatic accel/decel of feed motion (G01-G03) are in the exponential mode.

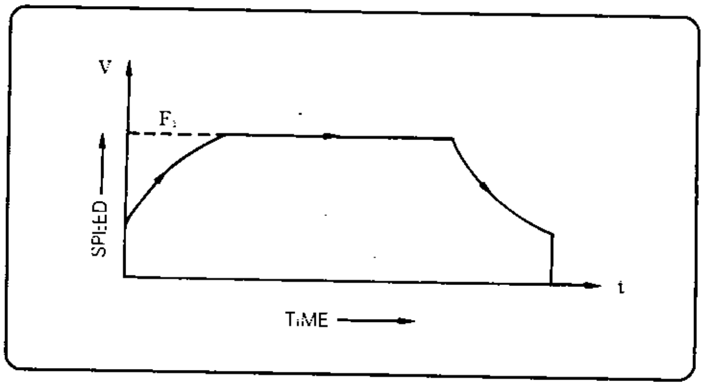


Fig. 2.5.13 Exponential Accel/decel Speed

(2) Feedrate time constants and feedreate bias are set by parameters. During tapping, time constants and bias other than for usual feedrate can be set by parameters.

(pm2501 to pm2505, pm2511 to pm2515, pm2821 to pm2828, pm2831 to pm2838)

NOTE
The automatic accel/decel parameters are set to the optimum values for the respective machines. Do not change the setting unless this is required for special purposes.

2.5.8 Override

For details on override, refer to command or reference manuals issued by your machine tool manufacturer.

2.5.8.1 Cutting feed override

(1) In the automatic operation mode (TAPE, MEM, or MDI), the feedrate instructed by the F or E code can be overridden in 21 steps in increments of 10% from 0 to 200%.

However, the feed by tapping cycle is as instructed by the F or E code, and cannot be overridden.

(2) If the OVERRIDE CANCEL switch is on, the feed is not affected by this selection switch and runs as instructed by the F or E code.

Table 2.5.18 Override

STEP	%	STEP	%
0	0	11	110
1	10	12	120
2	20	13	130
3	30	14	140
4	40	15	150
5	50	16	160
6	60	17	170
7	70	18	180
8	80	19	190
9	90	20	200
10	100	—	—

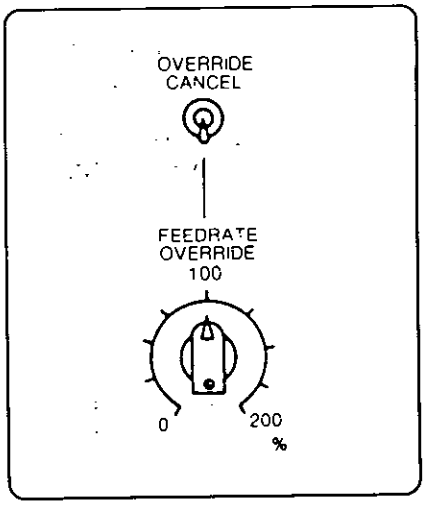


Fig. 2.5.14 Override Switch

(3) By option, the feedrate can be overridden in 32 steps in increments of 20% from 0 to 540%. In this case, it is overridden in increments of 20% from 220 to 30 0% and in increments of 40% from 340 to 540%.

Table 2.5.19 Override (Expanded Type)

STEP	%	STEP	%
22	220	28	380
23	230	29	420
24	240	30	460
25	250	31	500
26	260	32	540
27	270	—	—

NOTE Steps 1 to 21 are the same as those in Table 2.5.18.

2.5.8.2 Rapid traverse override

(1) The switch can be installed optionally to override rapid traverse rate.

(2) This switch is used to adjust the traverse rate by F_0, 25, 50, and 100%. F_0 is set by parameter pm2447.

By setting pm2000 D0=1, F_0 speed can be set in % for rapid traverse rate.

(3) The rapid traverse override is effective both in automatic operation (G00 command) and in manual operation (RAPID) mode).

NOTE Rapid traverse rate can be overridden in six steps by the parameters: 100%, 50%, F_0, F_1, and F_2.
F_1 and F_2 rates are set by parameters (pm2448, pm2449).
For details, refer to command or reference manuals issued by your machine tool manufacturer.

2.5.9 Dwell (G04)

This command can delay transition to the next block operation for the specified length of time.

2.5.9.1 Dwell per minute

```
G04  U(P, X, F)... ;
```

This command performs dwell for the length of time specified by the address U, P, X, or F.

Dwell is programmed as an independent block.

(1) Maximum value of time that can be specified by addresses U, P, X, and F

Table 2.5.20 Dwell Time
(P Specification Range)

Format	Dwell time (P programmable range)
U (P, X, F) 63	0 to 999999.999s

NOTE The value does not depend on metric/inch input or metric/inch output.

Example of programming

```
G04 U2.5;
```

Dwell time: 2.5 sec.

(2) Dwell is executed after pulse distribution in the preceding block enters the allowable value.

2.5.10 Speed Control Command

For speed control of interpolation, accel/decel control is executed by a time constant in start and stop. Therefore, part programs are created without regard to the control.

However, cutting in the corner may be rounded by follow-up error between NC command and actual machine movement. To eliminate the roundness, add G04 instruction in the corner for positive deceleration. For example, in the motion shown in **Fig. 2.5.15**, the tool moves as shown by the dotted line if X-axis starts moving before Z-axis is fully decelerated. To have the tool move along the solid line shown at the right of the figure instead of the dotted line, add G04 (dwell) in the corner portion.

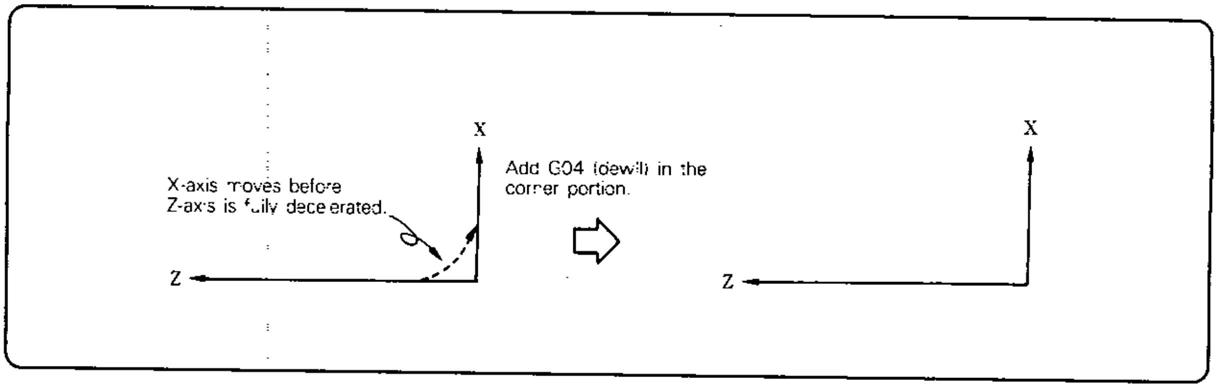


Fig. 2.5.15 Adding G04 (Dwell) in Corner

2.6 REFERENCE POINT RETURN

The reference point means a fixed position on the machine.

2.6.1 Automatic Reference Point Return (G28)

G28 X(U)...Z(W)... (*C(H)...*Y(V)...);

This command allows the machine to be returned to the reference point. The machine is automatically returned to the reference point after being positioned at the specified position by rapid feed.

This operation can be performed simultaneously up to two axes (* up to three axes) in one system. However, any axis for which coordinate command is omitted does not move.

Example of Programming

G28 X(U)...Z(W).... (C(H)...) ;

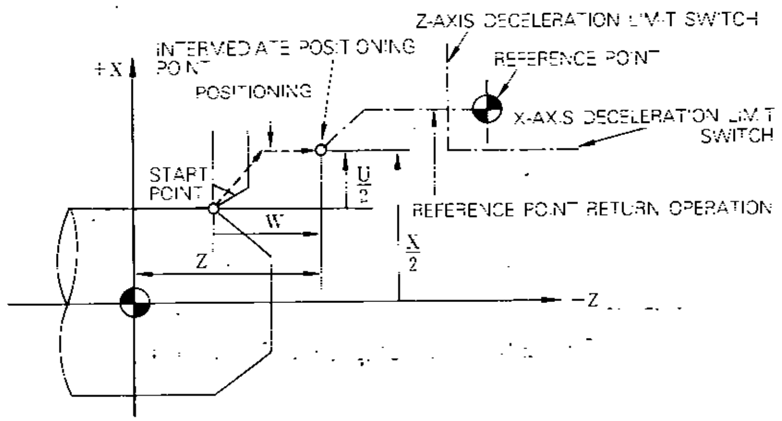


Fig. 2.6.1 Reference Point Return

(1) Reference point return operation

"Reference point return operation" means the same operation as a series of operations in which the movement is begun by manual reference point return operation and terminated when the machine returns to the reference point.

Reference point return is accomplished in two ways:

- Low-speed reference point return (a deceleration limit switch is used)
- Combination of low-speed reference point return the first time after power-on and high-speed reference point return the second and subsequent times (a deceleration limit switch is not used because the reference point is stored in the memory).

Thereafter, the former (a deceleration limit switch is used) is referred to as low-speed reference point return and the latter (a deceleration limit switch is not used) as high-speed reference point return.

NOTE C-axis* integrated with spindle of low-speed type does not use a deceleration limit switch.

(2) High-speed reference point return specification

High-speed reference point return (refer to pm4003 D6, D7) may be used in place of the automatic reference point return described above. In this case, movement is as shown below.

(a) Positioning directly at the reference point by rapid feed

After being positioned at immediate positioning point B, the machine is positioned directly at the reference point by rapid feed. This permits faster return to the reference point than in the reference point return operation in which the deceleration limit switch of each axis is used.

(b) Return from outside a reference point returnable area

Even if point B is located outside the reference point returnable area, the machine can be returned to the reference point.

(c) High-speed reference point returnable axis

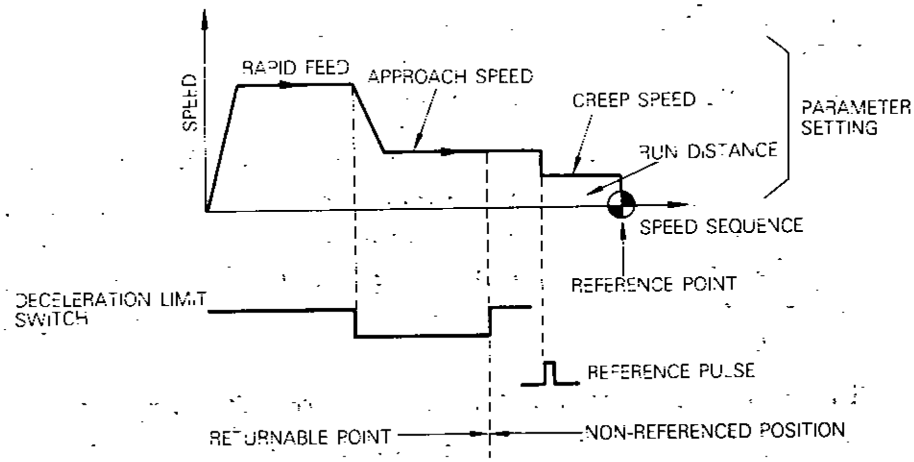
High-speed reference point return is only possible for the axes for which normal reference point return operation has been completed after power-on by manual reference point return or G28 command.

(d) When manual reference point return operation or normal reference point return operation has not been completed

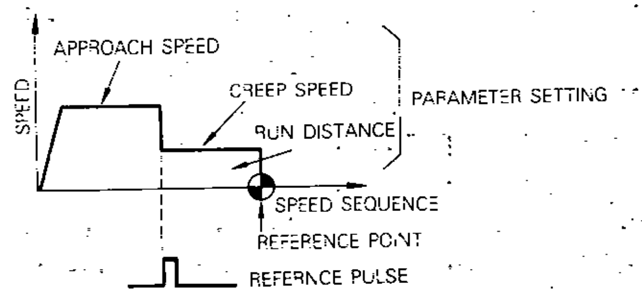
If normal reference point return operation has not been completed after power-on by manual reference point return or G28 command, normal reference point return operation is executed for X and Z axes instructed by G28.

(e) Automatic high-speed reference point return is valid for G28 command

Automatic high-speed reference point return is only valid for G28 command, and does not affect the operation of manual reference point return.



(a) X and Z Axes



(b) C-axis Integrated with Spindle

Fig. 2.6.2 Normal Reference Point Return Speed Sequence

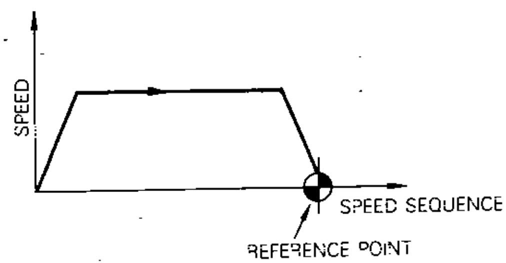


Fig. 2.6.3 Rapid Reference Point Return Speed Sequence

(3) C-axis* control integrated with spindle

When C-axis* control is performed integrally with the spindle, C-axis reference point return operation is performed for each switching from spindle mode to C-axis mode.

NOTES ON G28:

1. **Reference point return start position**

   The position in which reference point return operation can be started is as shown in Fig. 2.6.2. Alarm 2061 to 2068 will occur if reference point return from the return unable position is attempted to start. However, high-speed reference point return operation can be started in any position.

2. **When G28 is instructed**

   Before issuing G28, cancel tool position offset and nose radius compensation. Otherwise, they are automatically canceled.

3. **Reference point return valid/invalid**

   Reference point return valid/invalid can be selected for each axis. If reference point return invalid is instructed for an axis in the G28 block, alarm "241" results. Refer to parameter pm4022 D0-D7.

4. **Reference point return and move command**

   When any move command other than G28 is executed after power-on without performing reference point return, alarms "0411" to "0415" can be issued. Whether or not such alarms should be issued is specified by a parameter. For details, refer to parameters pm4004 D0 to D7.

   The direction of reference point return is set for each axis by pm4002 D0 to D7.

5. **Intermediate point**

   The absolute coordinate value of the axis instructed by the G28 block is stored in memory as an intermediate point. For axes not instructed by the G28 block, the intermediate point of G28 instructed before is stored in memory as the intermediate point for that axis.

6. **When M or T command is made in the same block as G28**

   If M or T command is made in the same block as G28, the machine is moved to the reference point regardless of whether FIN processing is completed or not until an intermediate point is positioned. Therefore, DEN output is made at the reference point.

7. **Reference point return and machine lock intervention**

   There are two types of machine lock intervened operations: machine lock is turned on after the machine is stopped by feed hold during movement, or the machine lock is turned off after the machine is stopped by feed hold once again.

   Table 2.6.1 shows how the machine is operated by machine lock intervention.

8. **Note on the first reference point return after power-on**

   For the first reference point return after power-on, attention should be paid to the position of the deceleration dog.

(CONT'D ON NEXT PAGE)

NOTES ON G28: (Cont'd)

Table 2.6.1 Movement by Machine Lock Intervention

		Machine lock intervention up to an intermediate point	Machine lock intervention during movement to reference point
Machine Lock OFF→ON	Low-speed type	Stopped after being moved to an intermediate point.	Display keeps moving continually to detect the operation of the deceleration LS. (The deceleration limit switch does not operate due to machine lock.)
	High-speed type		After machine lock is applied, the display moves to the reference point of the current value display (universal). (Without axial movement)
Machine Lock OFF→ON →OFF	Low-speed type	Machine moves to intermediate point. However, the position is deviated by machine lock intervention.	Moved up to the reference point. (Current value display is deviated by machine lock intervention.)
	High-speed type		The position is displaced by a quantity applied by the machine lock. Therefore, although the universal of the current value display becomes the reference point, the machine is not at the reference point.

2.6.2 Reference Point Return Check (G27)

This function checks whether a part program which is created to return to the reference point after starting from the machine reference point, correctly returns to the reference point.

G27 X(U)...Z(W)...Z(*C(H)... *Y(V)...) ;

When this command is given, the machine is positioned by rapid feed, at the instructed point simultaneously for two axes (*simultaneously for all axes), then it is checked whether that position is the reference point.

However, for axes for which coordinate command is omitted, positioning and check are not executed.

(1) **Operation after check**

If that position matches the reference point, the reference point return complete lamp lights. If all of the specified axes match the reference point, automatic operation is continued. If there is any axis for which the position is not matched, restore position error (alarm "0421" to "0425") results, and the automatic operation is interrupted. (The cycle start lamp goes out.)

(2) **G27 command**

If G27 is instructed during tool position offset, the machine is positioned at a point offset by the corrected quantity, so that the position does not match. Therefore, G27 can only be instructed after tool offset is canceled. Tool position offset cannot be cnaceled by G27 command.

(3) **Reference point**

The reference point means a point inherent to the machine to which the machine can be returned by "manual reference point return" or "G28 automatic reference point return."

(4) **Mirror image**

Mirror image is valid in the movement direction by G27 command. To avoid an unmatch error, be sure to specify G27 in the mirror image off mode.

(5) **No check is made if G27 is executed with machine lock.**

2.6.3 Return from Reference Point Return (G29)

This function causes the machine, after being moved to the reference point by automatic reference point return (G28 or G30) to be returned to the original position by going back on the same path as reference point return.

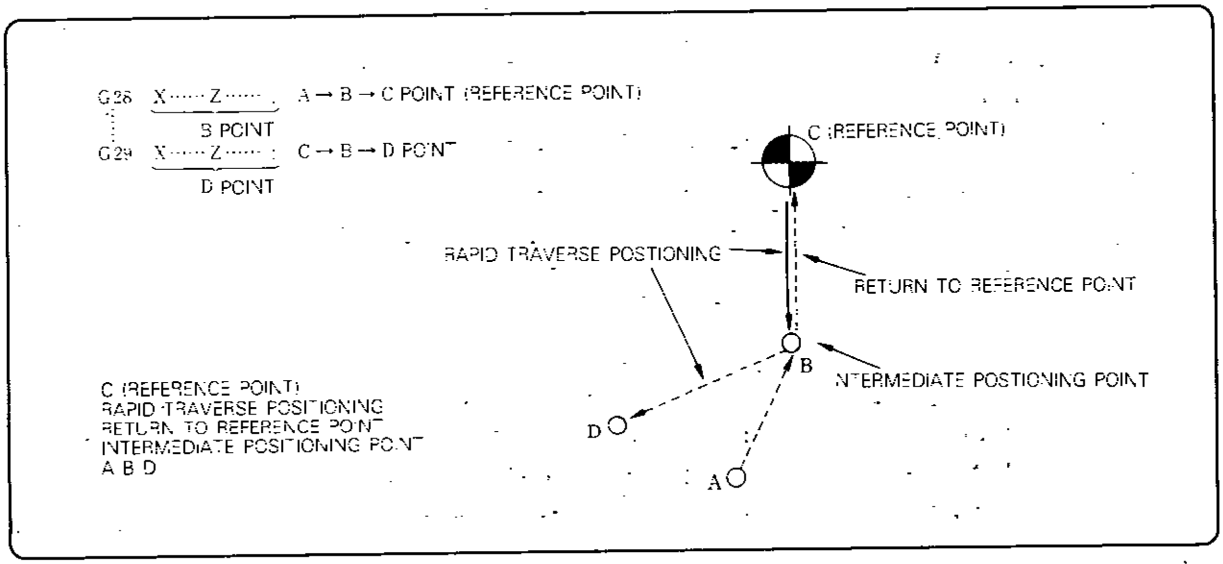


Fig. 2.6.4 Return from Reference Point Return

(1) Advantage of G29 command

If G29 is instructed, the distance between B and C need not necessarily be considered in the program. Especially, if an incremental command is used, G29 is useful in returning to the original coordiante system after once being returned to the reference point.

(2) Movement from C to D and from B to D

With G29, movement C → B and B → D is executed at rapid feed for two axes (✻up to three axes) simultaneously. However, any axes for which coordinate command is omitted do not move.

(3) If G28 or G30 (see Par. 2.6.4) is instructed a number of times, the coordinate value of point B for G29 movement becomes an intermediate point ultimately created by the latest G28 or G30.

(Example) Indicated by absolute command for easy understanding

		Intermediate point coordinate value X Z
N20 G28 X10. Z20.	→	(10. ,20)
N23 G28 X30.	→	(30. ,20.)
N24 G29 X−40. Z−50. ;		

G00 X30. Z20. ;
G00 X−40. Z−50. ;

Equivalent to these two blocks

Example of Programming

```
N31  T0300 ;
N32  G28  U80.  W20. ;
N33  T0400 ;
N34  G29  U-80.  W40. ;
  :
```

NOTE Incremental commands are effective for the command value of G29 (move quantity from an intermediate point) and the intermediate point is not affected.

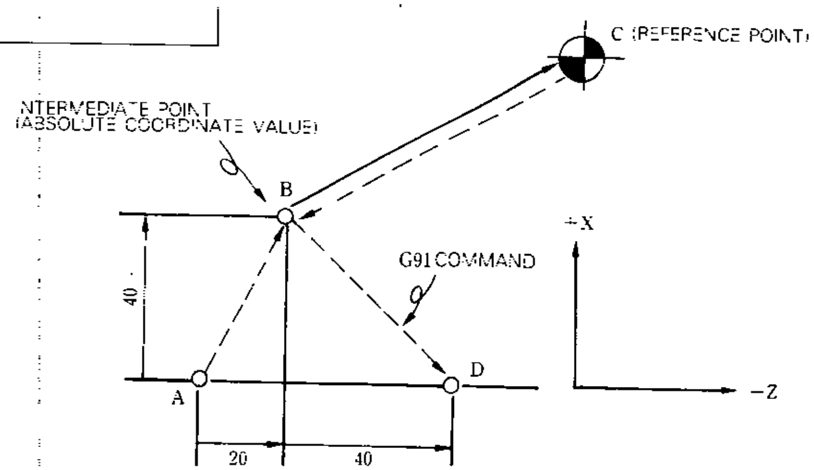


Fig. 2.6.5 B-Point Coordinate Value for G29 Operation

NOTES

1. **G29 command in nose radius compensation mode and canned cycle**

   If G29 is instructed in nose radius compensation mode (G41, G42) or canned cycle (G70 to G76, G90, G92, G94, G81 to G89), alarm "0170" or "0182" results.

2. **Note on G29 issued immediately after power on**

   If G29 is instructed after power is turned on without G28 or G30 being executed, alarm "0240" results.

3. **Note on issurance of G28, G30, or G29**

   As a rule, the tool position offset must be canceled before G28, G30, or G29 is instructed. If instructed while the offset is applied, intermediate positioning point B is also offset and the tool passes point B'.

   Tool position offset cannot be canceled by G29 command.

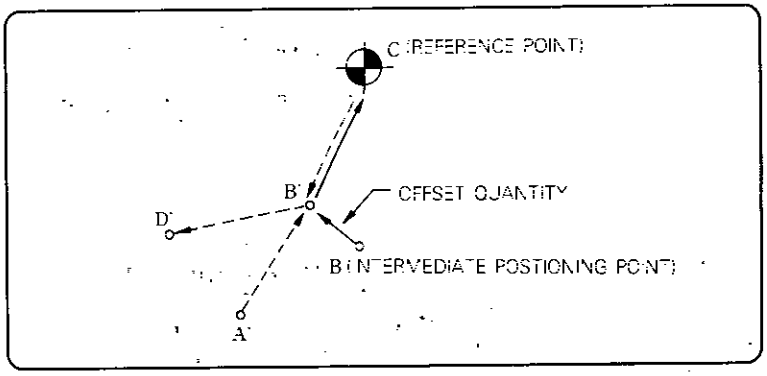


Fig. 2.6.6 Operation when Tool Position Offset is not Canceled

4. **Mismatch between the intermediate point of G28 or G30 and the intermediate point of G29**

   In the following cases, the intermediate point of G28 or G30 does not match the intermediate point of G29. Therefore, avoid making commands or operations that cause such mismatch.

   (a) When the following is performed from the time G28 is completed to when G29 is instructed
   - Setting of coordinate system (G50, key setup operation)
   - Intervention of machine lock
   - Intervention of manual operation in manual absolute OFF

   (b) When G28, G30, or G29 is instructed in a block after mirror image is canceled at a position different from the mirror image starting point

   (c) When G28, G30, or G29 is instructed after intervention of manual operation in manual absolute OFF.

2.6.4 Second to Fourth Reference Point return (G30) *

G30 Pn X(U)...Z(W)...(*C(H)...*Y(V))... ;

(Pn = P2, P3*, P4*)

This command causes the machine to be positioned at the second, third, or fourth reference point after being positioned at the instructed intermediate point.

- P2: Second reference point
- P3: Third reference point *
- P4: Fourth reference point *

If Pn is omitted, the second reference point is selected by default.

NOTE Any axe for which coordinate command is omitted does not move.

(1) Position of each reference point

The positions of each reference point are preset by parameters (listed below) as a distance from the first reference point.

- P2: pm6811 to pm6818
- P3: pm6821 to pm6828
- P4: pm6831 to pm6838

(Program example)

G30 P3 U−40. W30. ;

... The X and Z axes are returned to the third reference point.

NOTES ON G30:

1. **Note on G30 command**

   NOTES ON G3S in Par. 2.6.1 "Automatic reference point return (G28)" apply to the G30 command exactly as they are.

2. **G29 instructed after G30 is instructed**

   If G29 is instructed after G30 command, the machine is positioned at the point specified by G29 after passing an intermediate point specified by G30. However, the intermediate point is only updated for the axes for which G30 is instructed.

3. **Note on G30 command**

   Before G30 command is executed, manual reference point return or normal reference point return by G28 command must have been completed after power-on. If there is any axis in G30 command for which reference point return is not completed, alarm "0240" may result.

2.7 COORDINATE SYSTEM

2.7.1 Setting Coordinate System

Coordinate systems must be set before move commands can be programmed. Once a coordinate system is set, one absolute coordinate system is determined, so that all subsequently issued absolute move commands are moved along the set coordinate system.

(1) | G50 X.......Z.......(✻C....... ✻Y.......) ; |

With this command, the current tool position is set to the equipment as absolute coordinate point (X, Z, C).

In other words, the value of (X, Z, C) represents the distance (a signed value) from the desired coordinate origin (0, 0, 0) in the program to the current tool position. Therefore, the G30 command specifies the absolute coordinate origin.

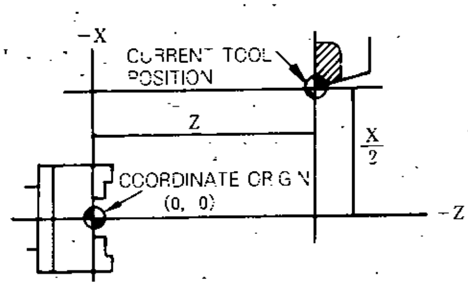


Fig. 2.7.1 Setting G50 Coordinate System at Origin Return

Fig. 2.7.1 shows an example of setting the G50 coordinate system at an origin return position, but G50 can be instructed at any position.

(2) | G50 U.......W.......(✻H....... ✻V.......) ; |

Setting of incremental G50

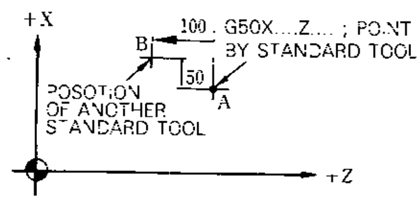
With this command, if address U, W, or H is specified, incremental values U (X-axis direction), W (Z-axis direction), and H (C-axis direction) are added to the established absolute coordinate system, thus establishing a new coordinate system.

For example, when there are tools of extremely different lengths, by dividing them into more than one tool group, then specifying the difference between standard tool position and another standard tool position by the incremental G50, one absolute coordinate system can be created.

Example of Programming

- Setting of incremental coordinate system

G50 U100. W−100. ;



(3) Coordinate system and tool position offset

G50 X80. Z62. ;

Use standard tool of tool number 01, execute G50 X80. X62.: for it to set a coordinate system, then select tool number 02 having the tool position offset quantity in **Fig. 2.7.2** to perform compensation, and the tool of number 02 will be moved to point A..

```
N 3  G50  X80. Z62. ;
N 4  G00  T0101 ;
       ⋮
N10  G00  T0202 ;
       ⋮
```

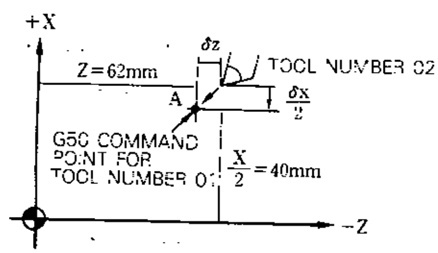


Fig. 2.7.2 Coordinate System and Tool Position Offset

By using the method of setting a coordinate system by standard tool and applying tool position offset to another tool as described above, tool movement can be programmed for all tool tips on one coordinate system.

NOTES ON G50:

1. **When T code is instructed in the next block in which G50 is instructed**

   When T code is instructed in the next block in which G50 is instructed, insert G00 as a rule. This is done to specify the speed of compensation movement by T code.

   ```
   G50  X.....Z..... ;
   G00  S500  M03  T0101;
   ```

2. **G50 is non-modal G code.**

   G50 is a non-modal G code valid only in the block in which it is command.
   As a rule, other G codes, and M, S and T codes cannot be instructed in an identical block.
   Especially, `G50 S..... ;` or `G50 T..... ;` has different functions and does not set a coordinate system.

3. **Note on G50 command**

   G50 must, as a rule, be instructed after canceling tool position offset and nose radius compensation.

4. **Set a coordinate system during operation.**

   The current position of a tool is set to coordinates (0, 0, 0) when power is turned on. Before operation, be sure to set a coordinate system.
   For C-axis integrated with a spindle, when the mode is switched to the C-axis mode and reference point return is completed, a coordinate system is automatically set.

5. **Current position of tool in G50 coordinate system**

   The current position of a tool G50 coordinate system is displayed in the current value field of the current value display screen.

6. **Resetting coordinate system**

   The coordinate systems are not affected by reset operation. To reset a coordinate system, perform any of the following:

   a. Perform key setup operation in coordinate system setting.
   b. Write "G50 X0 Z0 (C0);" command in MDI mode and execute it.
   c. Turn power on again.

7. **Validity for work coordinate system shift**

   When setting G50 coordinate system, whether work coordinate system shift is valid or not can be specified by the parameter pm4012 D0.

2.7.2 Automatic Setting of Coordinate System

A coordinate system can be automatically set on completion of manual reference point return.

For each axis, input beforehand the setting values at metric input to parameters pm4801 to pm4808 and the setting values of inch input to parameters pm4811 to pm4818.

(1) Validity of automatic setting of coordinate system

Whether automatic setting of a coordinate system is enabled can be specified by parameters pm4006 D0 to D7 for each axis.

(2) Use of work coordinate system shift function

The use of the work coordinate system shift function requires that a work coordinate system shift quantity be added to the coordinate value to be specified during automatic setting of a coordinate system.

(3) Coordinate system automatically set

In a coordinate system automatically set, functions enabled in another coordinate system such as G50 are disregarded.

2.7.3 Shifting Work Coordinate System

A coordinate system set by G50 and work coordinate system setting can be shifted by any distance.

In the same way as writing tool offset quantities, shift quantities can be written to work coordinate system shift memory (up to three pairs, one pair for each system) with offset memory number of "00" for each of X, Z, and (C) axes.

(1) Written shift quantity

A written shift quantity is enabled at the following moment:

(a) Execution of G50 coordinate setting
(b) Execution of G50T work coordinate system setting
(c) Execution of automatic setting of coordinate system
(d) Operations for coordinate system setting (key setup) of position and current value

In other works, during setting of each coordinate system described above, shift quantities are simply added to set the coordinate system. The tool is moved.

When the values of written shift quantities ΔX, ΔZ, and ΔC are positive, the coordinate system is shifted in the direction shown in **Fig. 2.7.3**. X_0 and Z_0 indicate the setting values in the original coordinate system.

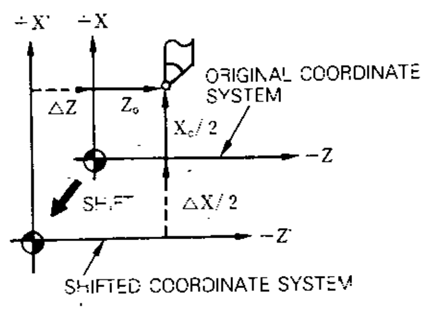


Fig. 2.7.3 Work Coordinate System

This function is subsequently executed at each moment of (a), (b), (c), (d), or (e) described above.

NOTE A shift direction can be reversed by parameter pm4012 D3.

(2) When the contents of work coordinate system shift memory are updated

When the contents of work coordinate system shift memory are updated, new shift quantities are made valid when one of the next a, b, c, d, or e is executed.

NOTES

1. Canceling work coordinate system shift

Shifting of a work coordinate system cannot be canceled only by specifying the value zero; it cannot be canceled by reset operation.

2. Tool offset number 01 specification and the contents of work coordinate system shift memory

T□□00: —— Canceling of tool position offset
G50T □□00: —— Setting of work coordinate system

The "00" specification in these commands has no relation to the contents of work coordinate system shift memory.

3. G50 execution and reset operation

During execution of G50 in (a) of (1) or reset operation in (d), whether shift quantities are valid can be specified by parameter pm4012 D0.

2.7.4 Setting Work Coordinate System (G50T, G51)*

When tools are replaced in the proper position for each tool, by setting a coordinate system by G501 in that position for each tool, one work coordinate system with a point on the workpiece as an origin can be saved. That is, programmers can perform programming in the work coordinate system through all processes.

(1) Tool coordinate value memory (number)

Before G50T is instructed, coordinate data for each tool must be written beforehand to the tool coordinate memory.

(a) Number of available tool coordinate memories

The number of available tool coordinate memories corresponds to the number of tool offset memory pairs. (**Table 2.7.1**)

Table 2.7.1 Number of Available Tool Coordinate Memories

	Number of tool offset memory pairs	Available tool coordinate memories (number)
1	0 to 16 per system	51 to 66 (16)
2	0 to 50 per system	51 to 99 (49)

(b) Tool friction memory number and tool number

Tool coordinate memory number "51" and tool number "01" correspond to each other. In the same way, memory number "52" and tool number "02" correspond to each other, and so forth.

Table 2.7.2 Tool Coordinate Memories and Tool Numbers

Tool coordinate memory	Tool number
51 ←	→ 01
52 ←	→ 02
⋮	⋮
80 ←	→ 30

(c) Coordinate data: Xtn, Ztn

Assume that coordinate data Xtn and Ztn shown in **Fig. 2.7.4** are written to tool coordinate value memory for each tool Tn.

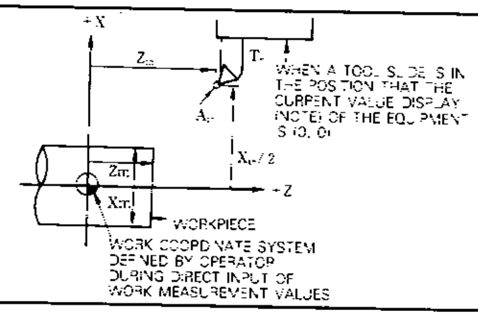


Fig. 2.7.4 Tool Coordinate Value Memory

(2) Setting of work coordinate system (G50T)

G50T □□ △△ ;

- △△ — Tool offset number specification (00 to 50)
- □□ — Tool coordinate memory number specification (51 to 99)

This command sets a coordinate system by finding a work coordinate system setting value by the following expression for each of X and Z axes:

> Work coordinate setting value = [value of current value indicator of equipment] (**NOTE**) [contents of instructed tool coordinate memory] [contents of instructed tool offset memory]

NOTE The value of current value indicator of equipment refers to the value displayed on the external current value screen in current value display (POS).

(a) Specifying tool offset number

Normally, specify "00" in the tool offset number specification field (△△).

(Example) G50 T51<u>00</u> ;

When "00" is specified, a work coordinate system is set with the contents of tool offset memory set to "0."

When the program of the above example is executed when a tool slide is in a given position [(—x, —y) by the current value display of the equipment, for example], the operator-defined work coordinate system is set correctly.

The reason is shown in **Fig. 2.7.5**.

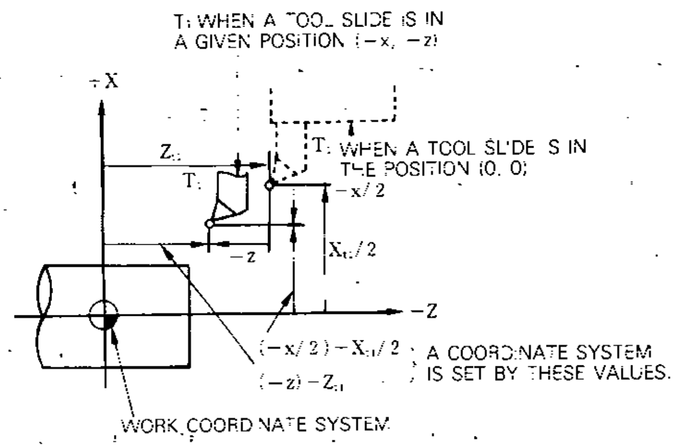


Fig. 2.7.5 Setting of Work Coordinate System

(b) G50 T0000 ;

This command sets a coordinate system by the value of the current value display of the equipment. Namely, the work coordinate system is canceled.

This is because a coordinate system is set by determining the following values from the T0000 specification to find a work coordinate setting value:

Tool coordinate memory = "0"
Contents of tool offset memory = "0"

(3) Current value display origin return (G51)

(a) G51 ;

This command positions a tool at the point of the current value display (0, 0) at rapid feed for both X and Z axes.

(b) **Processing at the position of the current value display (0, 0)**

Processing programs using a work coordinate system, as a rule, start processing at the position of the current value display (0, 0)

It is useful to use G51 to return to the starting point (0, 0) after completion of processing.

G51 must always be used in a single block.

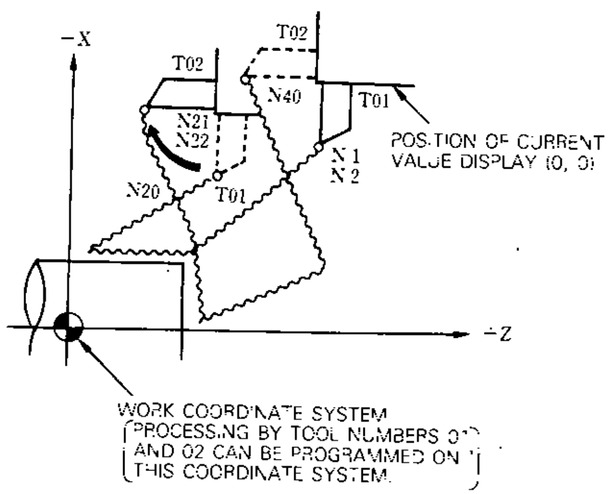
Example of Programming

• Example of use of work coordinate system

[A starting point is in the position of the current value display (0, 0).]

```
N1   G50  T5100 ;            ← Work coordinate system setting for tool No. 01
N2   G00  T0101  M03  S100 ; ← Selection of tool No. 01 (NOTE)
     :
 Work coordinate system setting
 for tool No. 1
     :
N20  G00  X……Z…… ;          ← Positioning at specified point
N21  G50  T5200 ;            ← Work coordinate system setting for tool No. 2
N22  G00  T0202 ;            ← Selection of tool No. 02 (NOTE)
     :
 Work coordinate system setting
 for tool No. 2
     :
N40  G51 ;                   ← Return to the point of the current
                               value display (0, 0)
```

NOTE Tool position offsets in T01<u>01</u> and T02<u>02</u> commands can be used as compensation for tool wear. They can also be used for compensation of tape processing during processing.

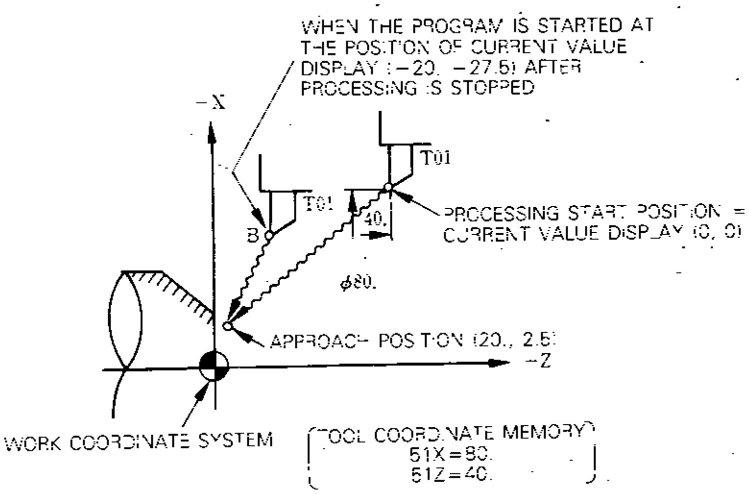


- Even when processing is stopped in the middle of processing beginning in the program shown below and the program is started from the beginning without returning the tool to the starting point of processing, the tool is positioned correctly toward the first approach position.

Example of Programming

- **Behavior when processing in a work coordinate system is stopped**

```
N 1   G50   T5100 ;
N 2   T0101 ;
N 3   G96   S150   M03 ;
N 4   G00   X20.   Z2.5 ;   ←─Ⓐ
        ⋮
```



Reason

Since the N1 G50 T5100; command sets a coordinate system using the following values, a work coordinate system is saved:

$$\begin{cases} X = (-20.)+(80.) = 60. \\ Z = (-27.5)+(40.) = 12.5 \end{cases}$$

Threrfore, the approach position A is unchanged.

Example of Programming

• When tool replacement positions in a work coordinate system are different

An example of programming and work coordinate system setting values when tool replacement positions are different are shown below.

Tool coordinate memory

No.	X	Z
51	100.	47.5
52	110.	40.

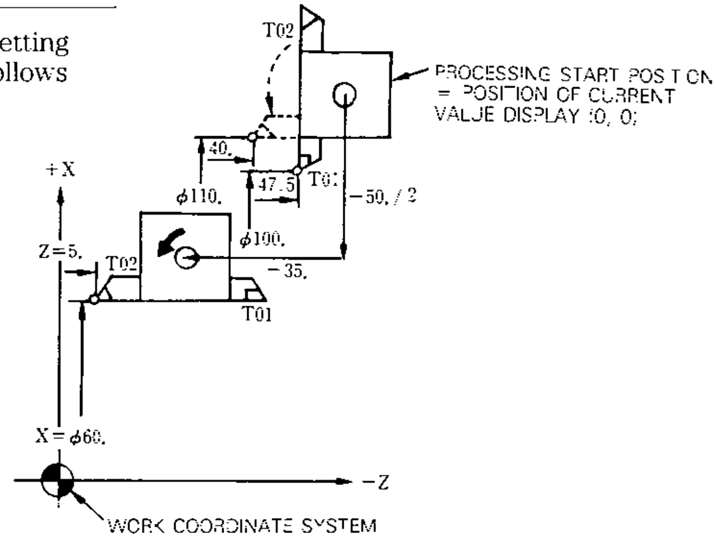
```
N 1   G50   T5100 ;
N 2   G00   T0101   M03   S1000 ;
            :
   (Processing by T01)
            :
N25   G50   T0000 ;
N26   G00   X-50.   Z-35. ;
N27   G50   T5200 ;
N28   G00   T0202   M03   S800 ;
            :
   (Processing by T02)
            :
N48   G51 ;
```

← N26: The tool replacement position of T02 is at the point (−50, −35) indicated by the current value display.

↳ N27: The coordinate system setting values are assigned as follows by this command:

X=(−50.)+110.=60.

Z=(−35.)+40. =5



NOTES ON G50T and G51:

1. G50T and G51 are non-modal G codes effective only in instructed blocks.
2. Notes on use of this function

   This function must be used with parameter pm3000 D0 = 0 (turn off the G50 preset of external current value display).
3. Note on $\boxed{\text{G51 ;}}$ command

   The $\boxed{\text{G51 ;}}$ command is equivalent to the following two block commands:

   ```
   G50  T0000 ;
   G00  X0  Z0 ;
   ```

   Therefore, after this command is executed, the tool offset number is canceled along with the work coordinate system, and the tool offset number is set to "00."
4. Work coordinate system shift

   The work coordinate system shift described in Par. 2.7.3 becomes effective when G50T work coordinate system setting is performed.
5. Displaying current position of tool

   The current position of a tool in a set work coordinate system is displayed in the current value display work coordinate system.

   It is not displayed in the external current value.
6. The work coordinate system by G50T is not canceled by reset operation.

2.8 ENTERING COORDINATE VALUES

2.8.1 Absolute and Incremental Commands

Either absolute coordinate commands or incremental move commands can be used by distinguishing between the use of address words X, Z, ✱C, and ✱Y, and the use of address words U, W, ✱H, and ✱V.

(1) Absolute commands use addresses X, Z, and C.

(Example) X.......Z.......C....... ;

(2) Incremental commands use addresses U, W, and H.

(Example) U.......W.......H....... ;

(3) Absolute commands and incremental commands can be used at the same time in an identical block.

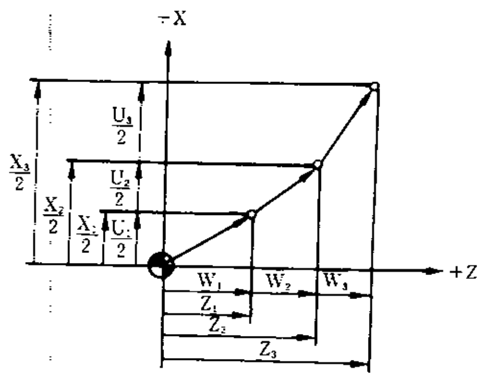
(Example) X.......W....... ;

U.......Z....... ;

NOTE When commands are issued simultaneously as in X.....U..... ; the command issued later becomes effective.

Table 2.8.1 Absolute and Incremental Commands and their Meanings

Address	Command value		Meaning (remarks)
X	Absolute	Diameter	Position in X-axis direction
Z			Position in Z-axis direction
✱C			Position in C-axis direction
✱Y			Position in Y-axis direction
U	Incremental	Diameter	Move quantity in X-axis direction
W			Move quantity in Z-axis direction
✱H			Move quantity in C-axis direction
✱V			Move quantity in Y-axis direction
I	Incremental	Radius	X-axis component from circular arc start point to circular arc central point
K			Z-axis component from circular arc start point to circular arc central point
✱J			Y-axis component from circular arc start point to circular arc central point
R	Incremental		Direct specification of circular arc radius



X, Z ..... Absolute commands
U, W .... Incremental commands

NOTE Since commands by X and U specify diameter values, actual move quantity is one-half of the value.

Fig. 2.8.1 Absolute Coordinate Values and Incremental Coordinate Values

Use of G90 and G91 (absolute and incremental commands)

(1) When special G code I (basic) or II (option) is selected, G90 and G91 codes can be used.

Table 2.8.2 Meanings of G90 and G91

G code	Meaning
G90	Absolute command
G91	Incremental command

NOTE G90 and G91 commands are effective only for addresses X, Z, and C as shown in Table 2.8.3.

Table 2.8.3 Effective Addresses of G90 and G91 Commands

Mode	Address	G90 command	G91 command
TAPE, MEM, MDI mode	X, Z, C, Y	Absolute	Incremental
	U, W, H, V	Incremental	Incremental

(Example) G91 G00 X40. Z50. ; .... Incremental move command

(2) Subdata I, K, and R of circular interpolation are always incremental commands.

NOTES

1. **Note on specification of addresses I, K, J, and R**

   Be sure to specify adresses I, K, J, and R specifying the center of circular arc with incremental values.

2. **G90 and G91 in identical block**

   G90 and G91 cannot be instructed in an identical block. When both of them are specified, the G code instructed later becomes effective.

   (Example) G01 G90 X80. G91 Z60. ;

   G91 becomes effective and both X and Z axes are placed in an incremental command block.

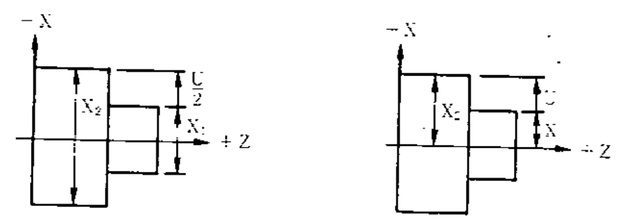
2.8.2 Selection of Diameter Specification and Radius Specification of X-axis.

Coordinate word addresses X and U are usually specified as diameter dimensions. This method is called diameter specification.

For X-axis, both diameter specification and radius specification can be used.

Which of them is used is specified by parameter pm1000 D1.

"0" .... Diameter specification
"1".... Radius specification



(a) Diameter specification (b) Radius specification

Fig. 2.8.2 Coordinate values

Table 2.8.4 Diameter Values and Radius Values

Item	Diameter specification	Radius specification
Address X Command	Diameter value	Radius value
Address U command	Incremental value of diameter	Incremental value of radius
X-axis Position	Diameter value	
Tool Position Offset Quantity	Diameter value	
Tool Coordinate Data for Tool Coordinate System	Diameter value	
Tip Radius Quantity	Radius value	
Feedrate F, E in X-axis Direction	Radius value per revolution Radius value per mm	
Radius Data I, K, J, and R of Circular Interpolation	Radius value	
G90 to G94, G70 to G76 Chamfering, Rounding, Compound Chamfering and Rounding Parameters D, I, K, P, Q, R	Radius value	

2.8.3 Inch/Metric Input Specification (G20, G21)

The G codes shown in **Table 2.8.5** specify the unit of input between metric or inch.

Table 2.8.5 Specification for Unit of Input

G code	Unit of input
G20	Input in inches
G21	Input in mm

The G20 and G21 codes are specified at the beginning of the program in an independent block.

When these G codes are executed, the following are matched to the altered unit of input.

- Subsequent programs
- Offset quantity
- Setting and part of parameters
- Part of mamual operations
- Various display

NOTES ON G20 and G21:

1. **Status of G20 and G21 at power on**

   The inch/metric selection setting is rewritten by G20 and G21. Therefore, the status of G20 and G21 at power on is determined by the setting parameter.

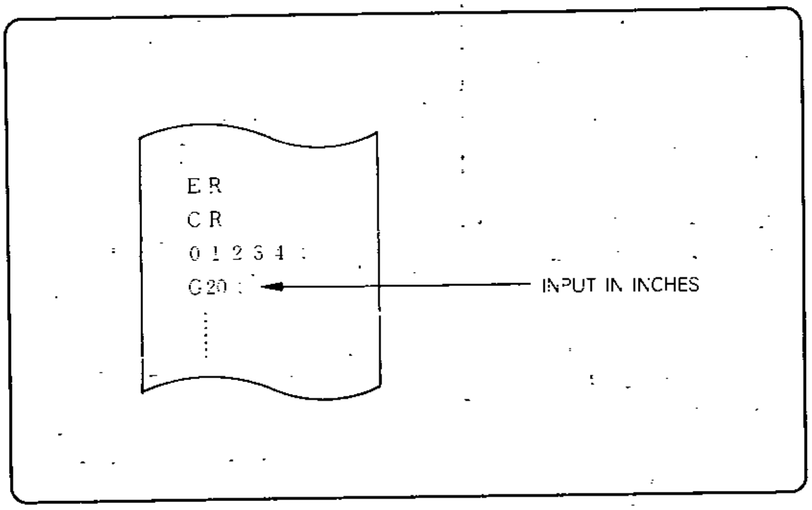


Fig. 2.8.3 Example of Programming

NOTES ON G20 and G21:

2. **Switching between G20 and G21 in the middle of a program**

   When selection between G20 and G21 is instructed in the middle of the program, cancel tool position offset and nose radius compensation.

3. **After switching between G20 and G21 is instructed, perform the following processing:**

   (a) Set coordinate systems before commanding a move command.

   (b) If work coordinate system display and external coordinate value display are to be used, clear the current value to zero.

4. **Tool offset quantity in G20 and G21 commands**

   The tool offset quantities stored in memory are handled differently in the G20 and G21 modes. Issue the G20/G21 command after correcting the offset quantities.

   Table 2.8.6 Tool Offset Quantities in G20 and G21 Commands

Stored offset quantity	G20 (inch) mode	G21 (inch) mode
15000	→ 1.5000 in	15.000 mm

5. **Inch input and metric input**

   In multiple systems, inch input and metric input are common to all systems. Inch input and metric input cannot be used independently in each system.

2.8.4 Decimal Point Input

Numeric values with decimal points can be used for addresses concerning coordinate word (distance), time, and speed.

The addresses in which decimal points can be used are as follows:

Coordinate word: X, Z, Y, C, U, W, V, H, I, K, R, A, and B

Time: U, P, and X

Feedrate: F and E

(Example)	[mm]	[Inch]
X15. →	X15.000 mm or	X15.000 inches
Z20.5 →	Z20.500 mm or	Z20.500 inches
(G90) F0.2 →	F0.200 mm/rev or (For F33)	F0.2000 inches/rev (For F34)
G04P1. →	Dwell 1.000s	

NOTE When numeric values without decimal points are input, they are normally handled by the equipment as 1 = 0.001 mm (or 0.0001 inch or 0.001 deg).

2.9 SPINDLE FUNCTION (S FUNCTION)

2.9.1 Spindle Command (S 5-Digit Specification)

The spindle speed (in units of r/min) can be directly specified by entering a 5-digit number following address S (S□□□□□).

If S-specification is instructed in the same block as M03 (spindle formal rotation) or M04 (spindle reverse rotation), the program normally advances to the next block after the spindle reaches the speed specified by S. For further information, refer to command or reference manuals issued by your machine tool manufacturer.

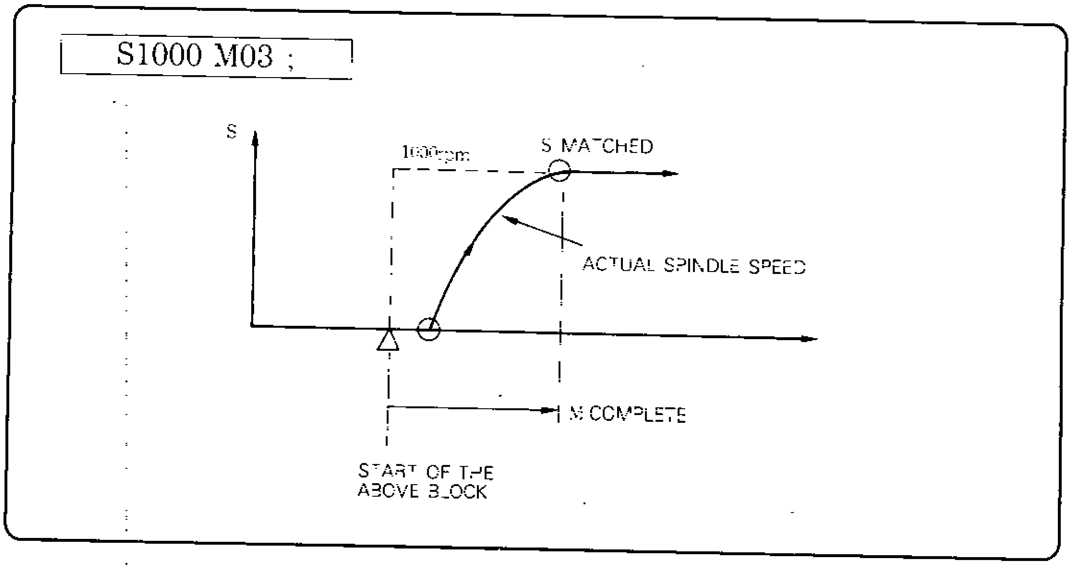


Fig. 2.9.1 Spindle Command

(1) **S-command is modal.**

The S-command is modal, so that once instructed, it remains valid until another S-command is given.

Even if M05 is instructed and the spindle actually stops, the S-command is retained. Consequently, the previously instructed S-command can be started by commanding M03 (or M04) again.

(2) **S 5-digit specification**

The control function in PC incorporating control equipment can be added to the output by S 5-digit specification.

In this case, the spindle rotation speed of manual operation can be set to a rotation speed equivalent to the S-command by the rotation switch installed in the machine pendant. For further information, refer to command or reference manuals issued by your machine tool manufacturer.

NOTES

1. **Changing S-commmand**

   If you change an S-command after the spindle is started by M03 or M04, pay attention to the selected range of spindle speed. For further information, refer to command or reference manuals issued by your machine tool manufacturer.

2. **Lower limit of S-command**

   The lower limit of S-command (S0 and its vicinity) is determined by the main axis motor and varies with each machine. For details, refer to command or reference manuals issued by your machine tool manufacturer.

   Minus S command causes an alarm "0102."

3. **Spindle speed override**

   Spindle speed override can be applied to an instructed S code.

4. **For machines that can switch spindle gear ratio by M code**

   Command beforehand which M code to select a desired gear ratio, then issue the S command. For details of gear ratio switching levels and the speed range of each gear ratio, refer to command or reference manuals issued by your machine tool manufacturer.

2.9.2 Spindle Maximum Rotation Speed Command (G50S)

G50 S....... ;

This command can specify directly the upper limit of spindle rotation in r/min with a five-digit number following the address S.

If S specification exceeding the upper limit is instructed after this command, the spindle rotation speed is clamped to the upper limit.

Peripheral speed-keeping control *

Under control to keep peripheral speed constant (G96 mode), when a too small X-axis current value for a specified peripheral speed (m/mm or ft/min) is given by a move command, the spindle rotation speed is clamped to the upper limit.

(Example) `G50 S2000 ;`

The spindle maximum rotation speed is clamped to 2000 r/min.

NOTES

1. The spindle maximum rotation speed specified in G50 can be viewed on the screen.
2. A specified maximum rotation speed is not affected by reset operation.
3. When the control function in PC is added to S output, the unit of address S is not always r/min. Refer to command or reference manuals issued by your machine tool manufacturer.
4. To release spindle clamp, issue `G50 S0 ;`

2.9.3 Peripheral Speed Keeping Control (G96, G97) *

This function uses the G codes shown in **Table 2.9.1**. These are modal G codes of 02 group. At power on, the spindle is in G97 (cancel) state.

Table 2.9.1 Meanings of G96 and G97 Codes

G code	Meaning
G96	Peripheral speed control specification
G97	Cancels peripheral speed control.

2.9.3.1 Peripheral speed keeping control (G96) *

(1) `G96 S.......(M03) ;`

This command specifies the peripheral speed of the workpiece with a numeric value of up to five digits following the address S. The units of peripheral speed are as shown in **Table 2.9.2.**

Table 2.9.2 Unit of Peripheral Speed

	Unit of peripheral speed
Metric input	m/min
Inch input	ft/min

If a peripheral speed is specified, the equipment regards the current value of X-axis as the diameter of the workpiece, finds the rotation speed of the spindle every 32 ms to keep the specified peripheral speed corresponding to changes in the current value, and outputs the result to analog voltage output. In subsequent blocks, the peripheral speed can be changed by S specification.

(2) **Switching spindle gear**

For machine tools having switchable spindle gears, insert the M code to specify gear switching before the block of G96 command. For details, refer to command or reference manuals issued by your machine tool manufacturer.

Example of Programming

```
      :
N8   M△△ ;                ← M code for switching spindle gears
N9   G96  S100  M03 ;       (Example:  Gear ratio No. 4 is specifed.)
      :
```

(3) G50 S specification

Specify spindle maximum rotation speed by G50 before the block in which G96 is specified. This is necessary to prevent abnormally high spindle rotation speed by peripheral speed keeping operation when the current value of X-axis becomes small.

```
N10  G50  S2000 ;   <-- The upper limit r/min value of spindle is specified.
N11  M△△ ;
N12  G96  S150  M03 ;
```

2.9.3.2 Cancelling peripheral speed keeping control (G97) *

```
G97  S.......(M03) ;
```

This command specifies directly a spindle rotation speed (r/min) with a numeric value of up to five digits following the address S. The peripheral speed keeping control is canceled and the spindle function by the S 5-digit specification is enabled again.

NOTES ON G96 and G97:

1. **Method for setting coordinate system**

   To perform peripheral speed keeping control, set G50 coordinate system or work coordinate system for programming so that X-axis coordinate value of spindle rotation center line is "0." That is, the X-axis coordinate value of a set coordinate system must represent correctly the diameter of a processing point of the workpiece.

2. **Treatment of tool position offset quantity**

   (1) To perform peripheral speed keeping control, generally set parameter pm4011 D5 = "0." This performs peripheral speed keeping control without adding tool position offset quantity to command coordinate values. Even when a large value is used as offset quantity, tool position offset is executed correctly as usual, and peripheral speed keeping control is also performed correctly.

   (2) When pm4011 D5 = "1" is set, the value of command coordinate value plus tool position offset quantity is regarded as the diameter of the workpiece and peripheral speed keeping operation is performed.

   Therefore, in this case, care should be taken so that a large value is not used as offset quantity by setting a coordinate system for each tool and using tool position offset as tool wear offset.

NOTE Switching of pm4011 D5 and pm300 D2 are independent from each other.

pm3000 D2 = "0" .... In current value display (work coordinate system), tool position offset quantity and nose radius compensation quantity are added to display the position.

pm3000 D2 = "1" .... In the same display as the above, tool position offset quantity is not added to display the position.

That is, switching of peripheral speed keeping control operation and switching to the current value display to CRT are performed independently from each other.

3. **Spindle gear switching**

   Up to four levels of spindle gear switching can be performed. **Fig. 2.9.2** shows major parameters required for this purpose. For details, refer to the parameter number list in Appendix.

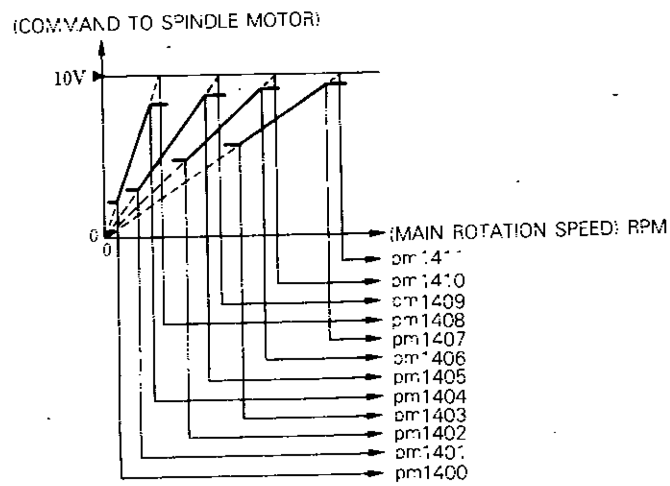


Fig. 2.9.2 Major Parameters

NOTES ON G96 and G97: (Cont'd)

4. **Peripheral speed-keeping control over positioning blocks**

In the case of parameter pm4011 D4 = "1", peripheral speed-keeping control is performed for positioning (G00, G06) blocks as well. For positioning, however, peripheral speed-keeping operation is performed for the end coordinates of the positioning blocks. Only cutting feed blocks are subjected to peripheral speed-keeping operation, as required.

In the case of pm4011 D4 = "0", peripheral speed-keeping control is performed only for cutting feed blocks and positioning blocks immediately before cutting feed. For positioning blocks, peripheral speed-keeping control is performed for their end coordinates.

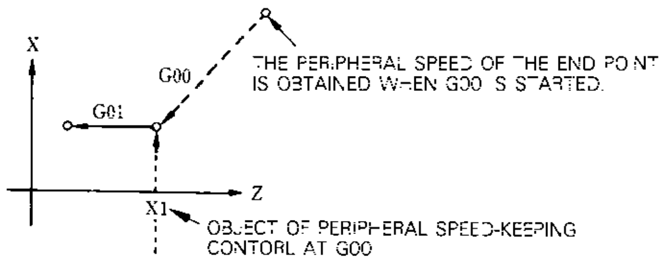
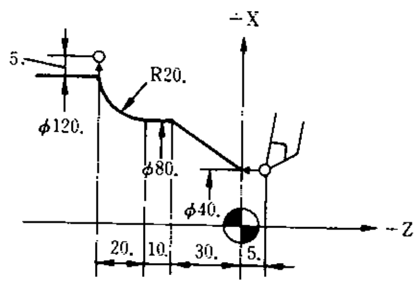


Fig. 2.9.3 Peripheral Speed-Keeping Control

```
N4   G50  S1500 ;      ← clamp of spindle upper limit r/min value
N5   M△△ ;             ← M code of gear switching
N6   G96  S150  M03 ;  ← Peripheral speed 150 m/min specification
N7   G00  X40.  Z5. ;
N8   G01  Z0  F0.15 ;
N9   X80.  Z-30. ;                 Under peripheral speed-
N10  W-10. ;                       keeping control
N11  G22  X120.  W-20.  R20. ;
N12  G01  U10. ;
N13  G97  S500 ;       ← Canceling of peripheral speed-
N14  G50  S2000 ;        keeping control
```



2.9.4 Rotary Tool Spindle Selection Function *

A rotary tool spindle can be attached additionally to the main spindle by option. In this case, the spindle rotation speed command switches the S command, using the G codes shown below.

Table 2.9.3 Rotary Tool Spindle Command G Code

G code	Meaning
G132	Designates subsequent S commands as rotation speed commands for rotary tool spindle.
G133	Designates subsequent S commands as rotation speed commands for main spindle.

NOTES

1. G132 and G133 are modal G codes.
2. G133 becomes effective at reset or power on.
3. When rotating the spindle after switching between G132 and G133, be sure to issue the S command again.
4. A rotary tool spindle cannot be used as a reference spindle for rotation feed.
5. The peripheral speed keeping function is ineffective for rotary tool spindles.

2.10 TOOL FUNCTION (T FUNCTION)

T functions have two functions: tool selection and tool offset.

2.10.1 T 4-digit Specification

A tool number and a tool offset number are specified with 4-digit numbers following the address T.

```
T □□ □□
  |   └─ Tool offset number specification
  └─ Tool selection specification
```

Specifiable offset numbers differ depending on their existence or absence, and the number of system. For details, refer to command or reference manuals issued by your machine tool manufacturer.

NOTES

1. When a command is input to change the tool selection specification program, a turret lathe generally starts locating the tool immediately. Therefore, issue the command after moving the machine to the position where the tool does not interfere.
2. The tool offset number 00 means canceling tool offset.
3. For details of tool offset, see Par. 2.12.

2.10.2 T 6-digit Specification *

A tool number and a tool offset number are specified with 6-digit numbers following the address T.

```
T □□□ □□□
   |    └─ Tool offset number specification
   └─ Tool selection specification
```

The T 6-digit specification is similar to the T 4-digit specification, except that the number of digits specified is increased.

2.11 MISCELLANEOUS FUNCTION (M FUNCTION)

2.11.1 M Function

The miscellaneous function is instructed by a number of up to three digits following M. Except in specific M codes, the definitions of M codes depend on the specifications of machine tool manufacturers. For details, refer to command or reference manuals issued by your machine tool manufacturer.

The following describes the specific M codes relating to the NC equipment.

NOTE M codes are marked with at power on. The M codes are not affected by reset operation.

2.11.1.1 M codes related to stopping (M00, M01, M02, M30)

(1) M00 (program stop)

When M00 is instructed during automatic operation, the automatic operation is interrupted after the operation of that block is executed and the M00R signal is selected. The operation can be restarted by depressing the cycle start switch.

(2) M01 (optional stop)

When M01 is instructed when the OPTIONAL STOP switch is on, the same operation as with M00 is performed. If the OPTIONAL STOP switch is off, M01 is disregarded.

(3) M02 (end of program)

M02 is used to terminate the program. When M02 is instructed during automatic operation, the automatic operation is terminated after the operation of that block is executed and then stopped. At this time, the NC equipment is generally reset. How the equipment actually responds varies with machine type. Refer to command or reference mamuals issued by your machine tool manufacturer.

(4) M30 (end of tape)

M30 is normally used to terminate the tape. When M30 is instructed during automatic operation, the automatic operation is terminated after the operation of that block is executed and then stopped. At this time, the NC equipment is generally reset and, simultaneously, memory rewind is executed. How the equipment actually responds varies with machine type. Refer to command or reference manuals issued by your machine tool manufacturer.

NOTES

1. When M00, M01, M02 or M30 is instructed, the NC equipment stops advance reading. For these M codes, M2-digit BIN code and individual decode signal are output.
2. Whether or not the spindle or coolant is turned off by M00, M01, M02, or M30 command depends on the manufacturer's machine specifications. For details, refer to command or reference manuals issued by your machine tool manufacturer.

2.11.1.2 Internal processing M codes

M codes from M90-M99 to M190-M199 are internally processed by the NC equipment. Even when these M codes are instructed, no external output signals (BIN code and decode outputs) are from the equipment.

*M92	Multiactive register OFF
*M93	Multiactive register ON
M96	Tool radius compensation C circular turnaround mode
M97	Tool radius compensation C intersection calculation mode
M98	Subprogram call
M99	Subprogram termination

M190-M199 are used for extension codes.

NOTE When multi-system control is performed, M codes of M800 to M999 may perform internal processing.

2.11.1.3 General other M codes

All M codes other than those listed as specific M codes shown in the preceding sections may be used for a variety of different purposes depending on the machine manufacturer's specifications.

Table 2.11.1 shows a typical usage example.

Table 2.11.1 Other Gemeral M Codes

M code	Meaning	Remarks
M03	Spindle forward rotation	Normally, M03 and M04 cannot be switched. Be sure to switch by using M05.
M04	Spindle reverse rotation	
M05	Spindle stop	
M08	Coolant ON	
M09	Coolant OFF	

When an M command is instructed in the same block as a move command, the following is determined depending on each machine.

(1) Whether or not the M command is executed simultaneously with the move command.

(2) Whether or not the M command is executed after the move command is completed.

Therefore, refer to command or refernce manuals issued by your machine tool manufacturer.

2.11.1.4 More than one M command in a block *

Up to five M commands can be instructed in one block, The instructed M codes and sampling data are output at the same time.

> **NOTE**
>
> There are limitations on M codes that can be instructed at the same time. For M codes that can be instructed at the same time, refer to command or reference manuals issued by your machine tool manufacturer.

2.12 TOOL OFFSET

- **Tool position offset**

  Shifts the tool position for the preset offset amount. Effective to the X and Z axes.

  The offset is enabled by the T command.

- **Nose R offset**

  Offsets the insufficient or excess infeed caused by the rounding of the tool nose.

  Enabled by G41, G42 and the nose R amount by the T command offset number.

- **Cutter radius offset** ✻

  Offsets the tool radius. Effective to polar coordinate interpolation and circular interpolation using the C-axis.

  Enabled by G41, G42 and the nose R amount by the T command offset number.

2.12.1 Offset Memory

The tool position offset, nose R offset and various offset functions as well as coordinate system setting data are stored in the offset memory.

The offset memory differs according to the option and serial number. Fig. 2.12.1 is the whole offset for 1 series.

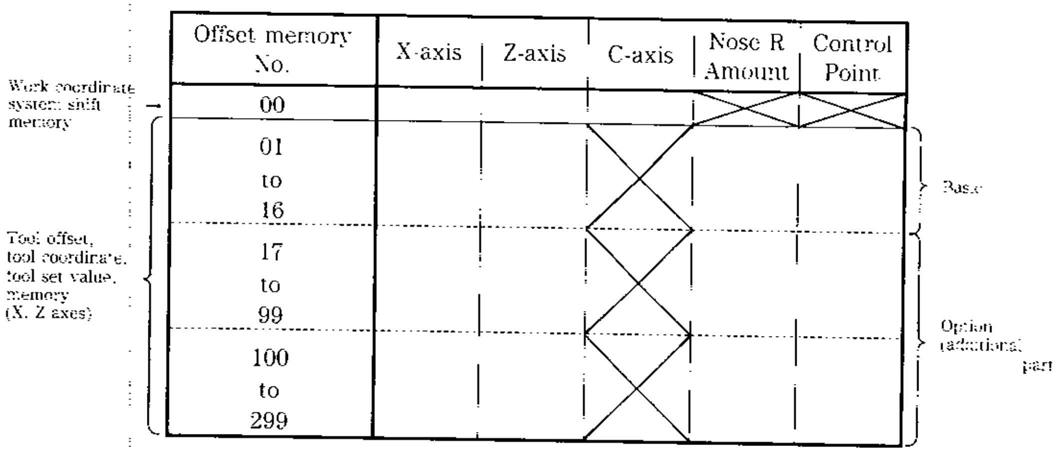


Fig. 2.12.1 Offset Momory (1 Series)

(1) X, Z axes offset memory of 001 to 299

The X, Z axes offset memory of 001 to 299 is generally used as the tool position offset amount, but can also be used as the tool coordinate value memory when using the work coordinate system. See the instruction manual of the machine tool builder for details.

(2) Multi-series control *

In multi-series control, the offset number can also be overlapped according to the series. See **Table 2.12.1**.

Table 2.12.1 Offset Number for Multi-series Control

Pairs	1 series	2 series *	3 series *
16	01 to 16	—	—
* 90	01 to 99	01 to 49 × 2	01 to 33 × 3
* 299	001 to 299	001 to 149 × 2 (01 to 99) × 2 **Note**	01 to 99 × 3

NOTE When T4-digit is specified offsets 01 to 99 can only be used, even when 299 sets of options are provided.

(3) "Tool offset number" specified by the T function

"Tool offset number" specified by the T function corresponds to the "offset memory number", and the data in the memory number are used for various offset procedures.

(Note that the tool coordinate memory number corresponds to the "tool selection" number in the T function. The work coordinate system shift memory is an individual function that has nothing to do with the T command.)

NOTE Store these offset data in the device before starting automatic operation.

NOTES
1. Both the 2-series and the 3-series control systems can have 2 offset series. (The control series and offset series are different.) 2. When using 2 or more offset series, the 16-set offset function cannot be used.

2.12.2 Tool Position Offset

In tool position offset, when a tool offset number is specified, the offset amount is added algebraically to the programmed coordinate command value, and the tool nose is moved to this corrected value. The difference between the coordinate value of the cutter nose and the coordinate value of the actual cutter nose can be preset in the tool offset memory as the offset amount.

In case the coordinate value of the tool nose changed by wear, the offset amount can be set again, so that machining as prescribed can be performed without changing the program.

(1) Setting range of the tool position offset amount

The tool position offset can be set within the following range.

Table 2.12.2 Toll Offset Setting Range

Output	Input	Tool Offset Setting Range
mm output	mm input	0 to ±9999.999 mm
	in input	0 to ±999.9999 in
in output	mm input	0 to ±9999.999 mm
	in input	0 to ±999.9999 in

(2) Tool position offset sign

The tool position offset amount is set as the difference of the X-axis and Z-axis (± value) from the reference tool position 0, into the offset amount memory. The sign of the tool position offset amount is the direction of the tool to be shifted.

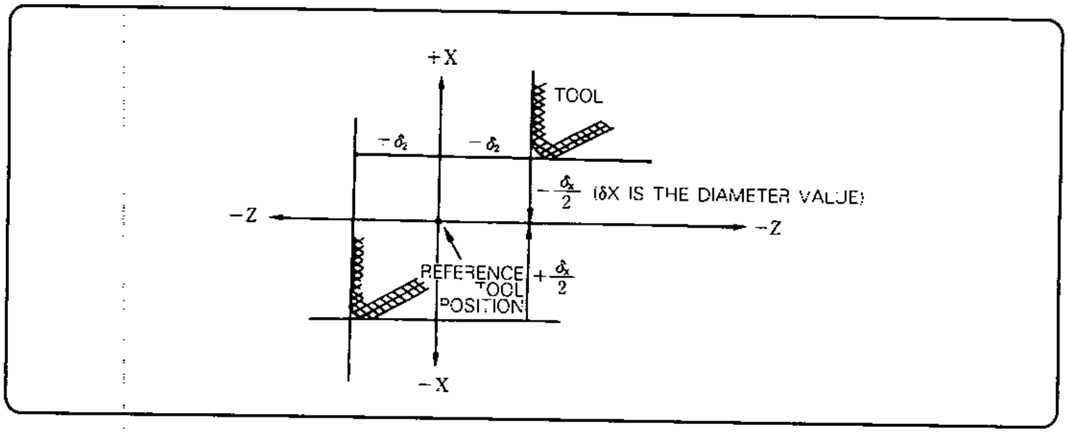


Fig. 2.12.2 Sign of Tool Position Offset Amount

(3) When the tool selected by T □□□□ moves according to the move command, the offset stored for the tool offset number is added algebraically to the command value, and moves the added point. If no move command is found in the same block, it moves off the offset position until another offset command is given. When another offset number is specified while an offset command is effective, or when the offset amount is changed, the difference is offset.

Program

```
T0101 ;  ←———— ①
   ⋮
G01 X…Z…F(E)… ;  ←———— ②
T0115 ;  ←———— ③
                  ⋮
```

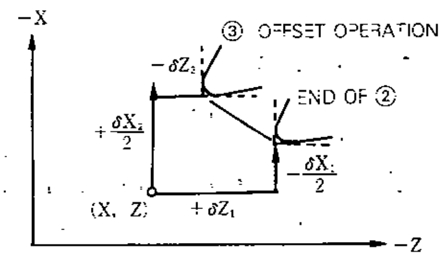


Fig. 2.12.3 Tool Position Offset Operation

(4) Offset move speed

The move speed of the above offset operation follows the current feed command. Therefore, when command tool position offset, the feed command (G00 or G01 plus F) is necessary in the block of the previous blocks.

(Example)

```
G50 X......Z...... ;
G00 S........M03  T0108 ;  ← Only offset is performed rapid
                             feedrate.
    X.......Z....... ;
    ⋮
```

(5) Commanding tool position offset

Tool position offset comes into effect when the tool offset number is specified.

(a) Commanding the T code

Command the T code in the block where the tool position offset is to be started.

The tool position offset is enabled immediately from the block. When the T code is read, the tool selection signal (binary code) is sent, and the offset starts by the offset amount specified by the offset number. T code is modal, and is maintained until another T code is specified.

(Example) G00 T0202 ;.....Tool number 02 is selected, and offset is performed according to the data in offset number 02.

(b) How to change the offset amount

The offset amount can be changed by commanding a new offset number.

(Example)

```
G00  T0202 ;
G01  X......Z......F....... ;
  :
G01  T0216 ;
```

(The T code is changed from 02 to 16, and offset is made with the cutting feedrate.)

NOTE Note that changing the tool selection in the above alerts the tool indexing operation.

(c) How to change the angle of taper cutting

The taper cutting angle can be changed by the following command. Command the T code that changes the offset number in the same block as the cutting feed command.

```
① → G00  T0202 ;
    G01  X……Z……F…… ;
         ⋮
② → G01  U+……W−……F……T0216 ;
```

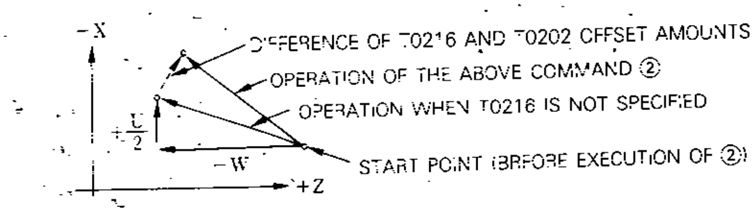


Fig. 2.12.4 Correcting the Tapering Angle

When the T function and move command are commanded in the same block, the tool nose moves to the target value after offset. Thus, in the above example, tapering is performed with the angle corrected for the difference between the T0202 and T0216 offset amounts.

(d) Canceling tool position offset

The tool position offset can be canceled by commanding T code with offset number 0 or 00 (T□□00). The tool position offset is canceled immediately in the same block.

(Example 1)

```
    G00  T0202 ;
    G01  X.......Z.......F....... ;
           :
    G01  U+.......W-.......F.......T0216 ;
           :
①  → G00  X.......Z.......T0200 ;
```

← Stops adding the offset amount, and moves to the X, Z axes position as commanded.

(Example 2) The above block 3 can be divided into two parts.

```
G00  X.......Z....... ;
     T0200 ;
```

← Only the move for cancel is performed in rapid feed.

NOTES

1. **Canceling tool position offset**

   The tool position offset can be canceled by the "reset operation".

2. **M02, M30 commands**

   Always command offset cancel before giving the M02, M30 commands.

3. **Automatic zero point return (G28)**

   Generally command offset cancel before commanding automatic zero point return (G28).

4. **Reset during tool offset**

   When "reset operation" is given or when the unit is reset by the M02, M30 commands during tool offset, the tool offset is canceled and the offset number designation changes to 0 (or 00).

5. **Tool offset at reference point return**

   When reference point return (automatic, manual) is executed, the tool offset is temporarily canceled. The later operation differs according to the following parameter.

   Parameter pm4010 D1.

   0: The offset data are maintained, and recover from the following block.
   1: The offset data are canceled.

6. **G27 (reference point return check) command**

   Always command the offset cancel before commanding the G27 (reference point return check) command. If G27 is commanded when tool offset is on, the offset amount is added to the coordinate command, to cause reference point return check fault.

2.12.3 Nose R Compensation (G40, G41/G42) *

Since the tool nose is round, tool position compensation is not enough for tapering or circular interpolation, causing insufficient infeed or excess infeed as shown in **Fig. 2.12.5**. Such error is compensated so that the cutting is performed correctly according to the machining program by the nose R compensation function.

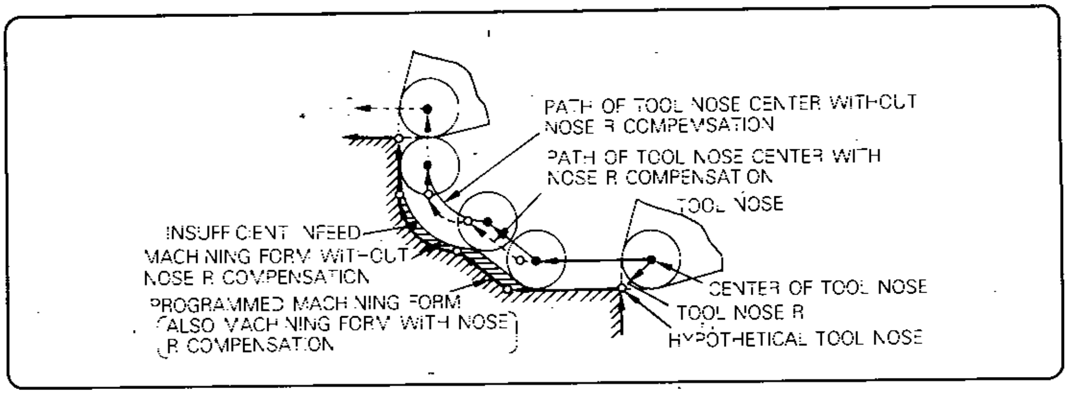


Fig. 2.12.5 Nose R Compensation

(1) Nose R amount

(a) Nose R amount memory

Before commanding nose R, the value of nose R of the tools used must be written into the nose R memory in the device.

The nose R amount memory set that can be written differs according to the machine.

(b) Range of nose R amount

The nose R amount can be set within the range as shown in **Table 2.12.3**.

Table 2.12.3 Nose R Amount Command Range

	Nose R amount Command Range
Metric Input	0 to 99.999 mm
Inch Input	0 to 9.9999 in

(c) Setting the nose R amount

1 The nose R amount is set by the radius of the tool nose circle, without sign. (**Fig. 2.12.6**).

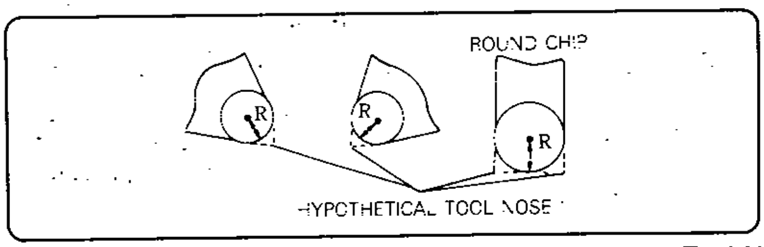


Fig. 2.12.6 Setting the Nose R Amount and Hypothetical Tool Nose

(d) Hypothetical tool nose

In nose R compensation, the hypothetical tool nose point is the reference position. The current position display of the NC also indicates the move of the hypothetical tool nose point.

The above nose R value can be written according to the instructions given in Par 2.3, "Displaying and writing the tool offset amount". Address R is used.

(2) Specifying the hypothetical tool nose position (control point)

(a) Control point memory

The position of the hypothetical tool nose viewed from the nose R center is represented by 1-digit numerals (0 to 9) which are called control points. As the nose R amount, the control point is to be previously written into the memory of the NC.

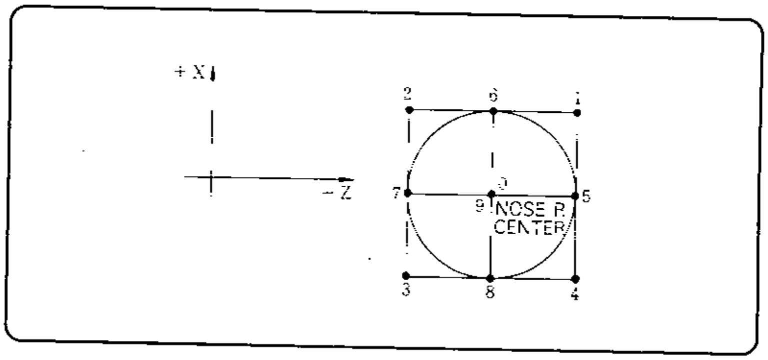


Fig. 2.17.7 Control Point

(b) Setting the control point

The control point is set by 1-digit numerals (0 to 9) indicating the hypothetical tool nose, as shown in **Fig. 2.12.7**.

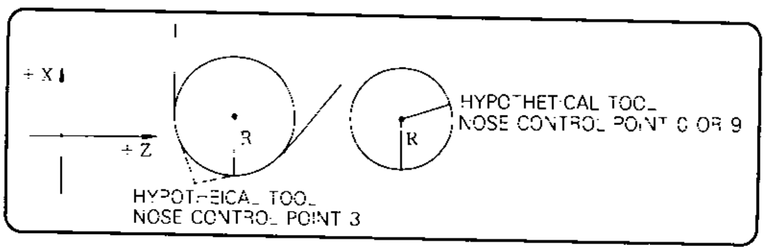


Fig. 2.12.8

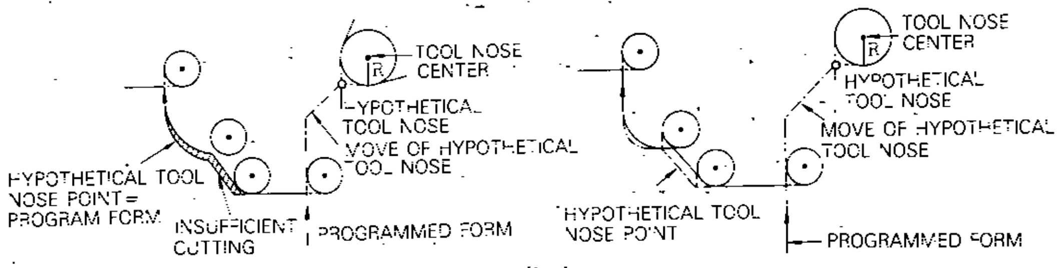
The control point value is written using address C.

NOTES
Control point 0 is the same as control point 9 when parameter pm4013 D6 = 1. When parameter pm4013 D6 = 0, nose R compensation is disabled.

(c) Control point and program

When using control points 1 to 8

Set the coordinate system with the hypothetical tool nose position as reference to create the program.



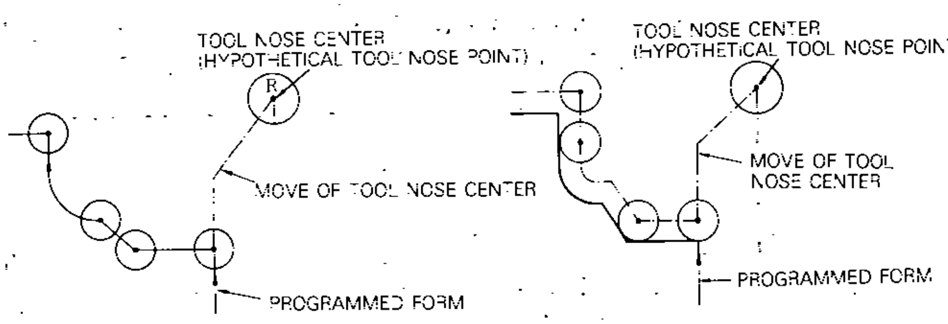
(a) Without nose R
The hypothetical tool nose point and the programmed form match, to leave uncut parts or over-cut parts.

(b) With nose R
Nose R is offset against the programmed form, to prevent uncut parts or over-cut parts.

Fig. 2.12.9 Programmed Move of Control Points 1 to 8

When using control point 0, 9

When using nose R, set the coordinate with the tool nose center as reference, and create the program. When not usiung nose R, the program form and finishing form must be different.



(a) Without nose R
The tool nose center and the programmed form match. Therefore, when the tool nose center is the refernce of the coordinate system, the programmed form must be different from the finishing form.

(b) With nose R
As Fig. 2.12.9 (with nose R), appropriate nose R compensation is commanded, to prevent uncut parts or over-cut parts.

Fig. 2.12.10 Programmed Move of Control Points 0 and 9

(3) Nose R compensation command

(a) Designation of offset amount, control points

The nose R compensation offset amount and control points are specified by commanding the offset memory number where the nose R compensation amount and control points are preset, into the lower 2 digits (or 3 digits) of the T command.

```
T □□ □□    (T4-digit designation)
 │    └─ Tool position offset and nose R
 │       compensation offset number,
 │       control point memory number
 │       designation
 └─ Tool selection designation
```

(b) Designating nose R compensation on, and direction of compensation

The nose R compensation on/off and offset direction are specified by the G codes as shown in **Table 2.12.4**.

Table 2.12.4 G Codes Used for Nose R Compensation

G Code	Description
G40	Cancel nose R compensation
G41	Compensation on left side of move direction (nose R center is at the left side)
G42	Compensation on right side of move direction (nose R center is at the right side)

G code that turns compensation ON (G41/G42)

Specify G41 or G42 with the T code to start nose R compensation. The G codes are also used to specify the direction of nose R compensation.

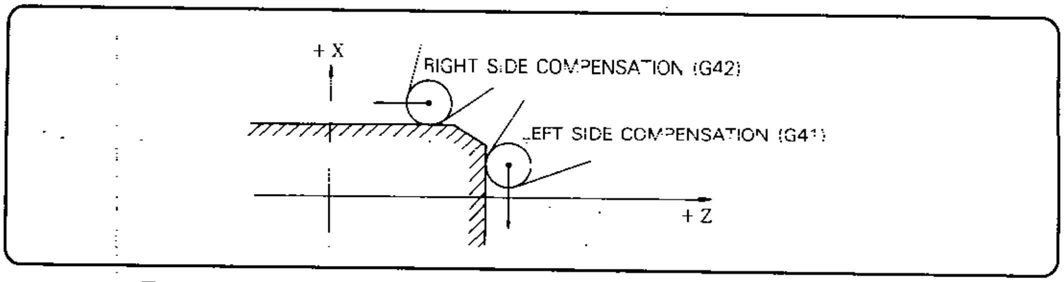


Fig. 2.12.11 Designation of Direction of Nose R Compensation

The direction of compensation can be switched from G42 to G41, or vice-versa in the middle of the program. It is not always necessary to cancel the nose R compensation by the T□□00 before switching the direction.

Specify G40 to cancel the nose R compensation

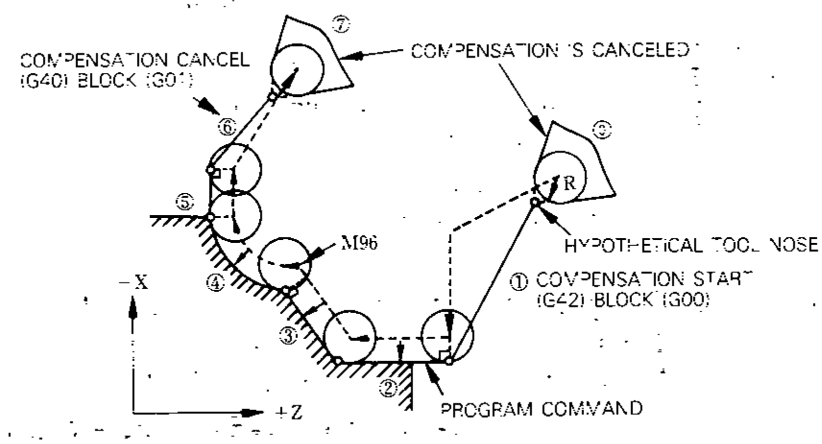
Specify G40 to cancel the nose R compensation.

NOTES

1. G40, G41, and G42 are group 06 modal G codes, which are effective until the next designation.
2. The compensation is canceled to G40 when the power is supplied or reset.

(4) Outline of the nose R compensation

Fig. 2.12.12 shows an outline of how nose R compensation works.



① When compensation is canceled, the programmed position and the hypothetical tool nose match. ⓪ and ⑦

② During compensation mode, the nose R center moves along a path nose R shifted from the programmed command. Therefore, the hypothetical tool nose point and the programmed position do not match. (The current value display shows the hypothetical tool nose point).

③ During compensation mode, two blocks are connected by the tool nose center path passing through the intersection (M97) or by forming an arc (M96). In the above figure, blocks 3 and 4 are connected by an arc path.

④ The compensation start block 1 and compensation cancel block 6 move in a straight line while switching between compensation and cancel. Special care must be taken for giving commands in these blocks.

Fig. 2.12.12 Nose R Compensation (G42, Control Point 3)

The nose R compensation is applied against the programmed form offset by tool position offset.

(5) How to enter the offset mode

(a) When tool offset number by both T code and G41 (G42 to G44) are commanded

It enters the offset mode when tool offset number by both T code and G41 (or G42 to G44) are commanded, to start nose R compensation. Strictly saying, offset mode starts when the T code and G code AND conditions are satisfied.

Therefore, either the T code or G code can be commanded first (**Fig. 2.12.13**). The move when it enters the offset mode from an offset cancel status, is called the start up movement.

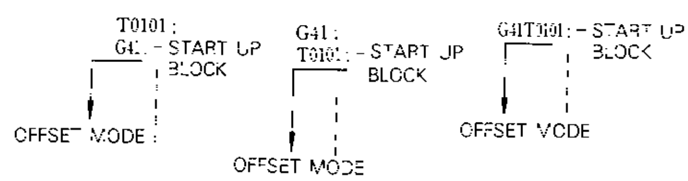


Fig. 2.12.13 How to Enter the Offset Mode

(b) Start up movement

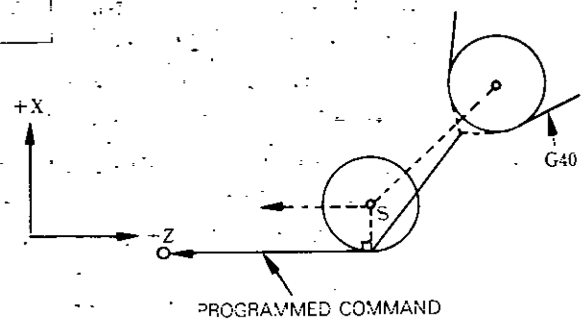
The start up movement differs according to whether there is a move command in the start up block or not.

When the start up block has move command

① **Inside corner (180 degrees or less) start up**

The tool nose center moves onto the normal of the direction (vector) next to the start up block.

```
T······ ;
G01  G42  Z······X······ ;
Z······ ;
```



NOTE "S" is the single-block stop point

Fig. 2.12.14 Inside Corner Start Up (i)

Circular command

```
T······ ;
G01  G42  Z······X······ ;
     G02  Z······X······R······ ;
```

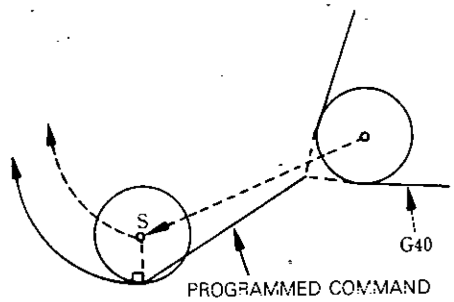


Fig. 2.12.15 Inside Corner Start Up (ii)

② Outside corner (180 degrees or more) start up

There are two types, switched by parameter.

Parameter pm4013 D0 = 1: type A
Parameter pm4013 D0 = 0: type B

[1] Type A ; The tool nose center moves onto the normal of the direction (vector) next to the start up block.

(A-i)

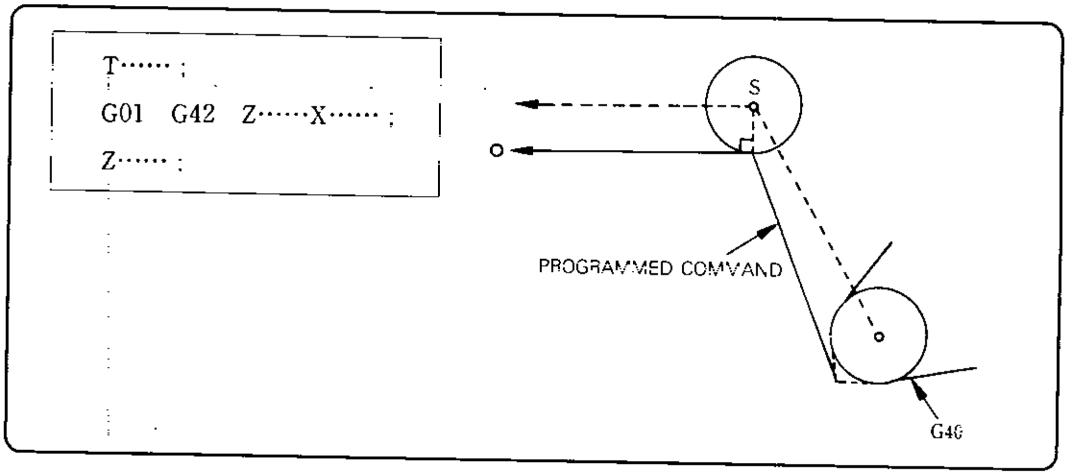


Fig. 2.12.16 Outside Corner Start Up (i)

(A-ii)

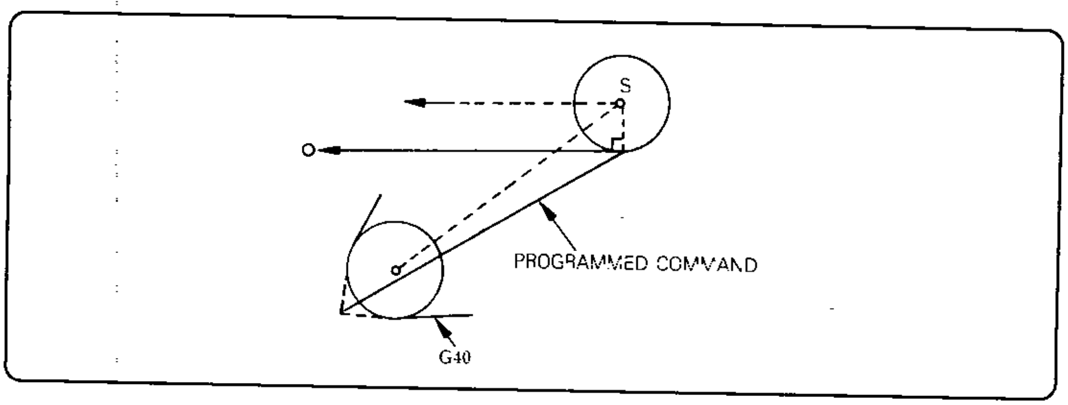


Fig. 2.12.17 Outside Corner Start Up (ii)

(A-iii)

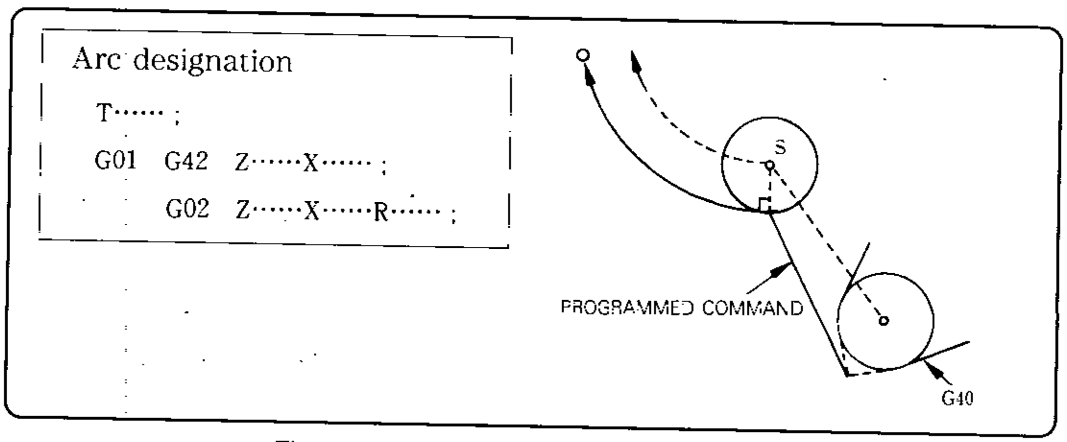


Fig. 2.12.18 Outside Corner Start up (iii)

[2] Type B : It starts up so that the tool does not cut into the block next to the start up block.

Outside corner (180° ≦ θ < 270 deg)

(B-i)

```
T······ ;
G01   G42   Z······X······ ;
Z······;
```

(M97 mode)

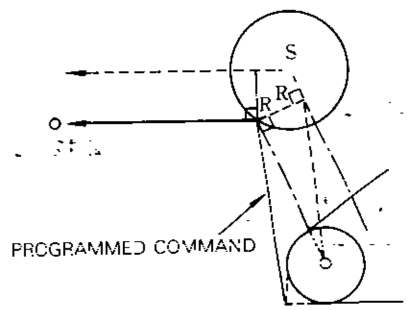


Fig. 2.12.19 Outside Corner Start Up (iv)

(M96 mode)

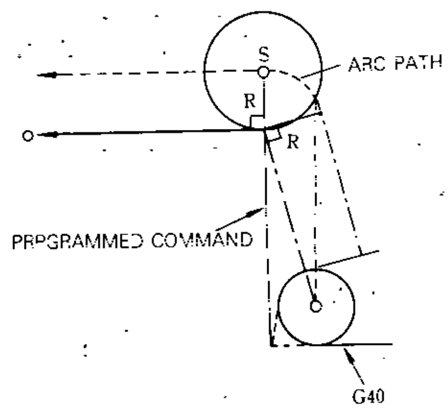


Fig. 2.12.20 Outside Corner Start Up (v)

(B-ii)

(M97 mode)

Arc command
T······ ;
G01 G42 Z······X······ ;
G02 Z······X······R······ ;

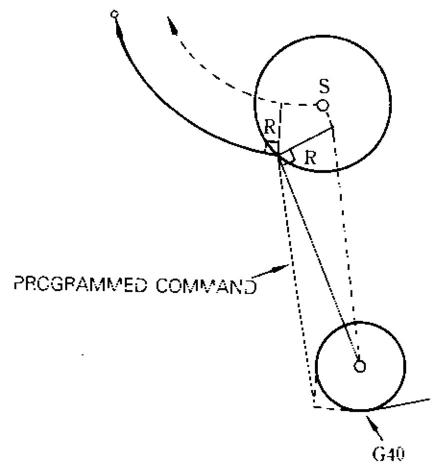


Fig. 2.12.21 Outside Corner Start Up (vi)

(M96 mode)

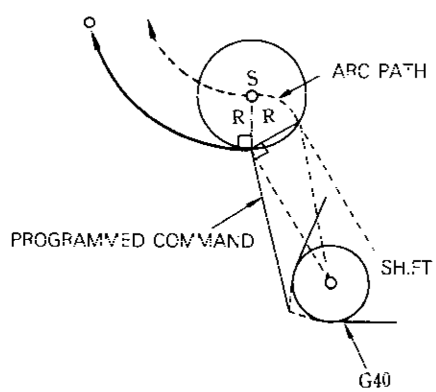


Fig. 2.12.22 External Rounding Start Up (vii)

Outside corner (270 ≧ 0 deg)

As in Fig. 2.12.20 and Fig. 2.12.22, when under the M96 mode, it moves around the end point of the start up block and the start point of the next block in the arc.

Under the M97 mode, the movement is as shown in Figs. 2.12.23 and 2.12.24.

(B-iii)

```
T…… ;
G01  G42  Z……X…… ;
Z…… ;
```

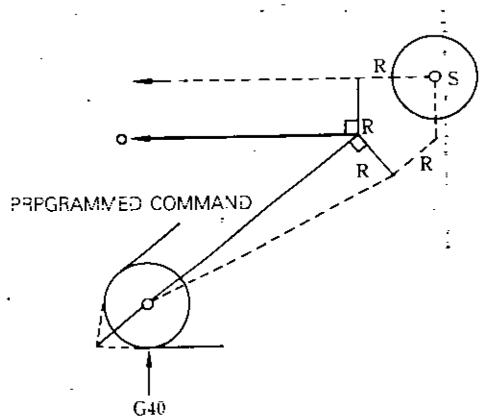


Fig. 2.12.23 External Rounding Start Up (viii)

(B-iv)

```
Arc command
  T…… ;
  G01  G42  Z……X…… ;
  G02  Z……X……R…… ;
```

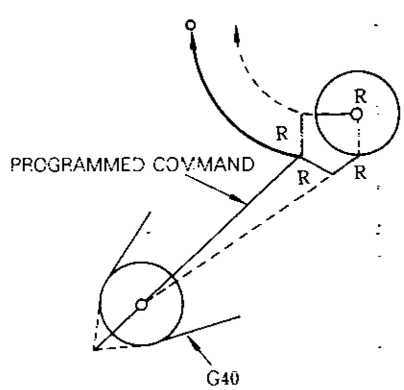


Fig. 2.12.24 External Rounding Start Up (ix)

When no move command is found in the start up block

The center of the nose R moves to the R-offset position of the normal of the start point of the block next to the start up block, regardless of inside corner, outside corner, M96, M97 status.

```
T······ ;
G01  G42  F······ ;
G01  Z······X······ ;
```

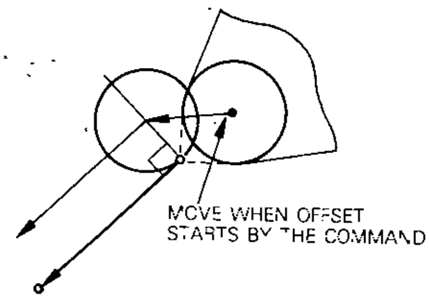


Fig. 2.12.25 Start Up when There is No Move Command in the Start Up Block (i)

Arc command

```
T······ ;
G01  G41  F······ ;
G03  Z······X······I······ ;
```

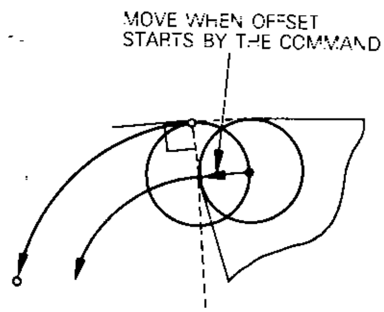


Fig. 2.12.26 Start Up when There is No Move Command in the Start Up Block (ii)

NOTES

1. **Group 01 G codes that can be commanded in the start up block.**

   G00, G01, G11 are the only group 01 G codes that can be commanded in the start up block. Commanding other G codes will cause alarm "0180".

2. **G11 start up**

   If G11 start up is initiated by the first cutting feed, chamfering is the same as the G01 command in the offset mode.

(6) Move under the offset mode

During the offset mode, the nose R center moves along a path offset for R. Calculation of the path is automatically made by the NC unit, so it is only necessary to specify the form of the graphic in the program. The tool path is controlled as follows according to the angle between the blocks.

(a) Normal movement

(i) Inside-corner (180 degrees or less) : Intersection operation

Line and line

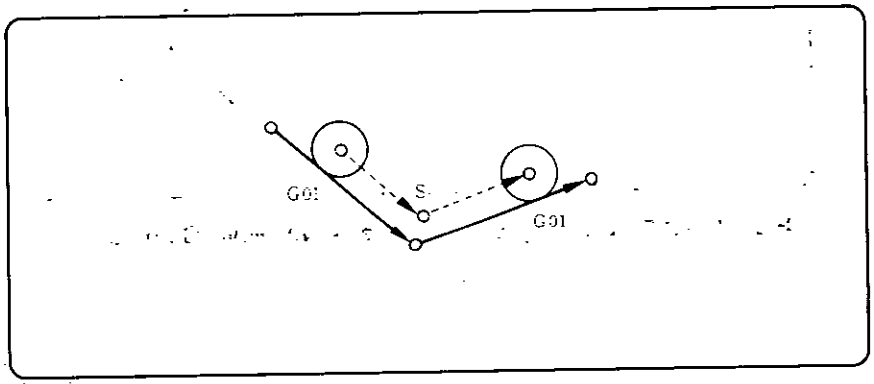


Fig. 2.12.17 Line and Line

Line and arc

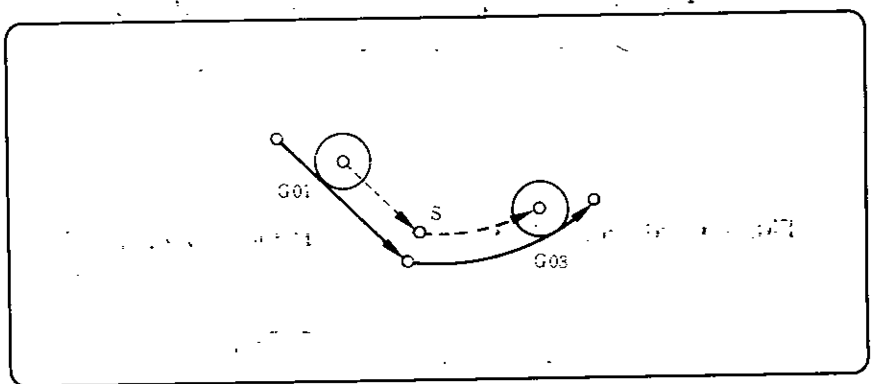


Fig. 2.12.28 Line and Arc

Arc and arc

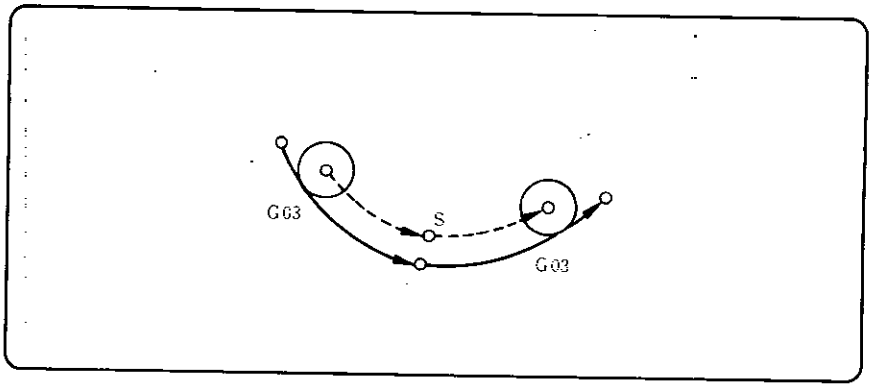


Fig. 2.12.29 Arc and Arc (ii)

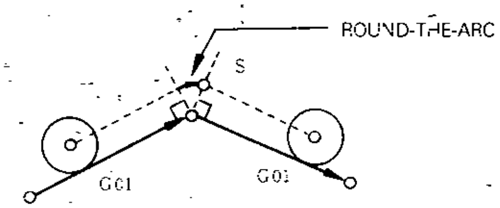
(ii) Outside corner (180 degrees or more)

There two types as follows, switched by the internal M code.

M96 ... Tool radius offset round-the-arc on M97 ... Tool radius offset round-the-arc off (Executes intersection operation)

① Round-the-arc ON

Line and line



NOTE The movement of round-the-arc is included in the preceding block

Fig. 2.12.30 Line and Line

Line and Arc

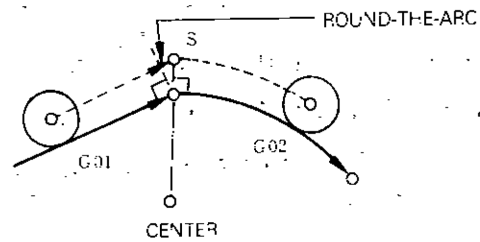


Fig. 2.12.31 Line and Arc

Arc and arc

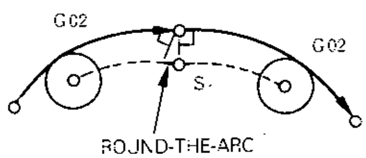


Fig. 2.12.32 Arc and Arc

② Round-the-arc OFF

- When material angle α is $90 \leqq \alpha < 180$ degrees

Line and line

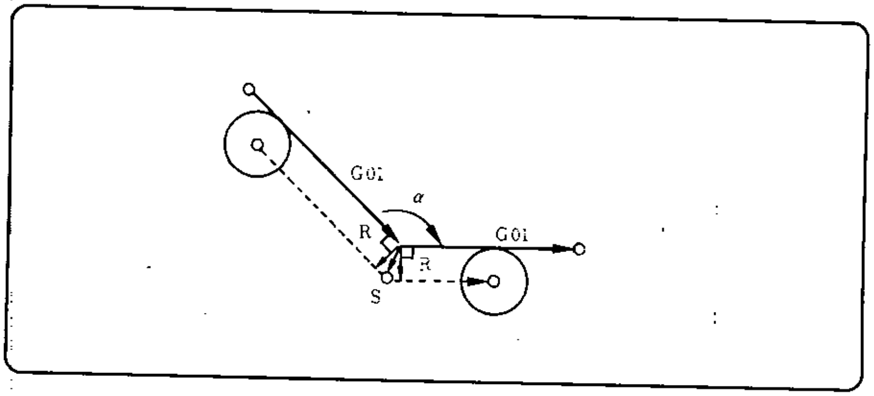


Fig. 2.12.33 Line and Line

Line and arc

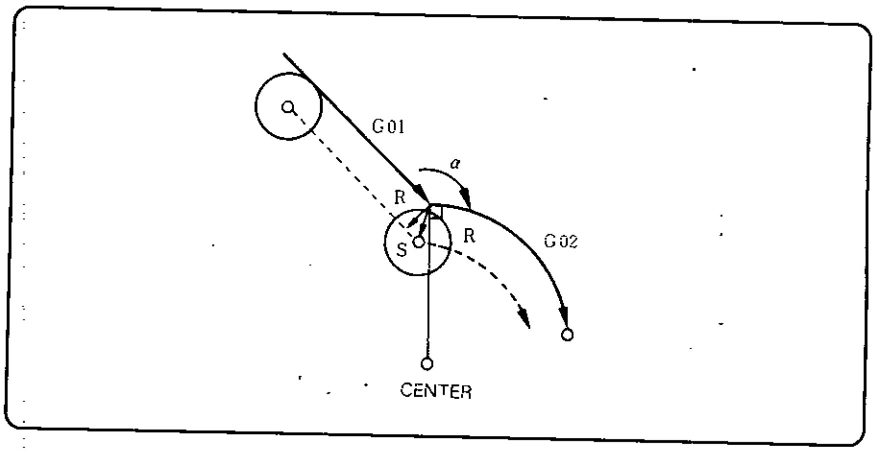


Fig. 2.12.34 Line and Arc

Arc and arc

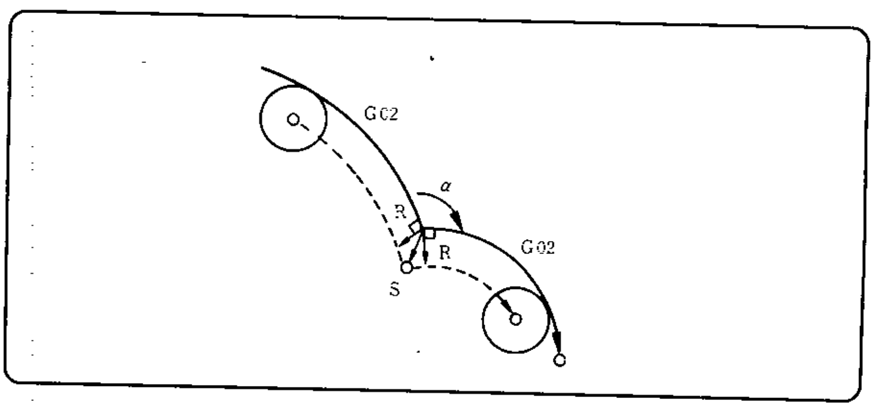


Fig. 2.12.35 Arc and Arc

- When material angle (α) is $\alpha < 90$ degrees

Line and line

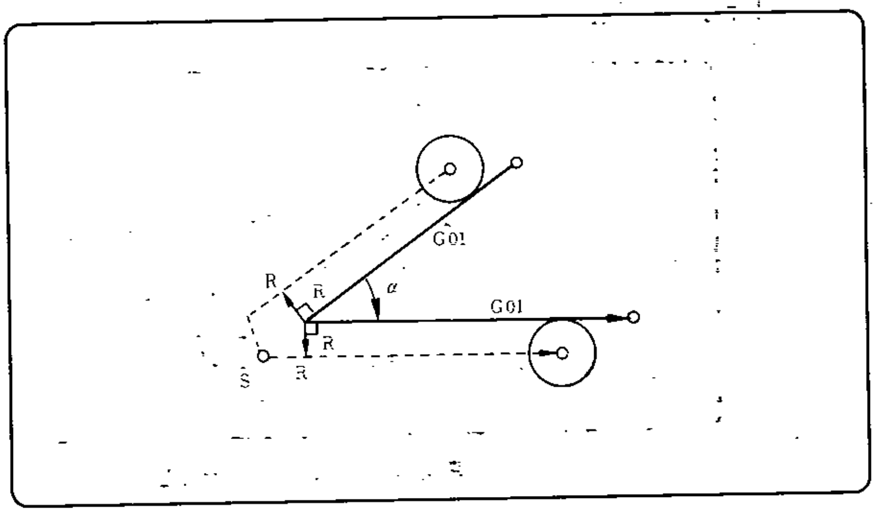


Fig. 2.12.36 Line and Line

Line and arc

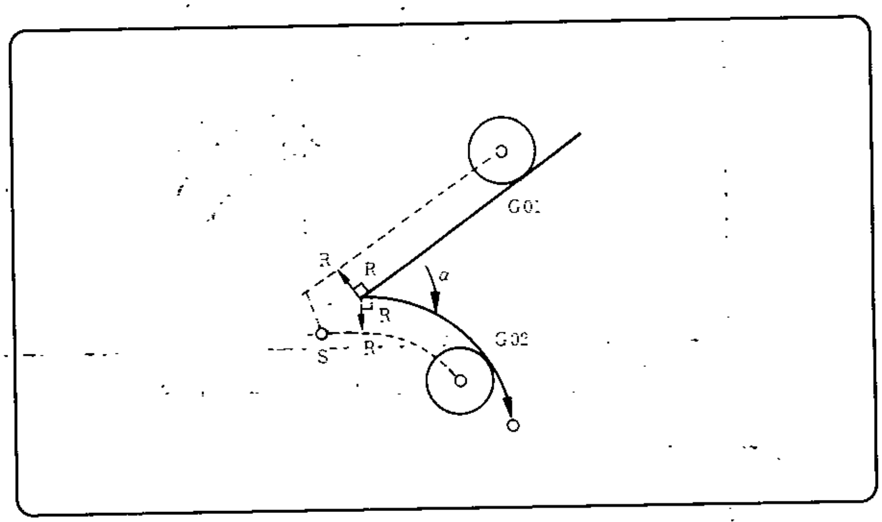


Fig. 2.12.37 Line and Arc

Arc and arc

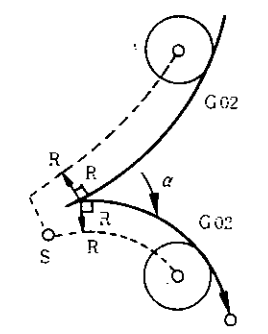


Fig. 2.12.38 Arc and Arc

(b) Special command, movement in offset mode

(i) Temporary cancel command (type I)

The movement is temporarily canceled by commanding as follows in the offset mode.

During the temporary cancel, the nose R center moves to the R-offset position from the end point of the previous block.

① When 3 consecutive blocks without move command are commanded

The following are non-move commands.

- G04 (dwell)
- M code single-command
- S code
- Move command with move amount = 0

② When look-ahead stop block is commanded

The following are look-ahead stop blocks.

- M00, M01, M02, M30
- Parameter-set look-ahead stop M codes
- G36 to G39 (stored stroke limit area on/off)
- G10 (tool offset amount setting)

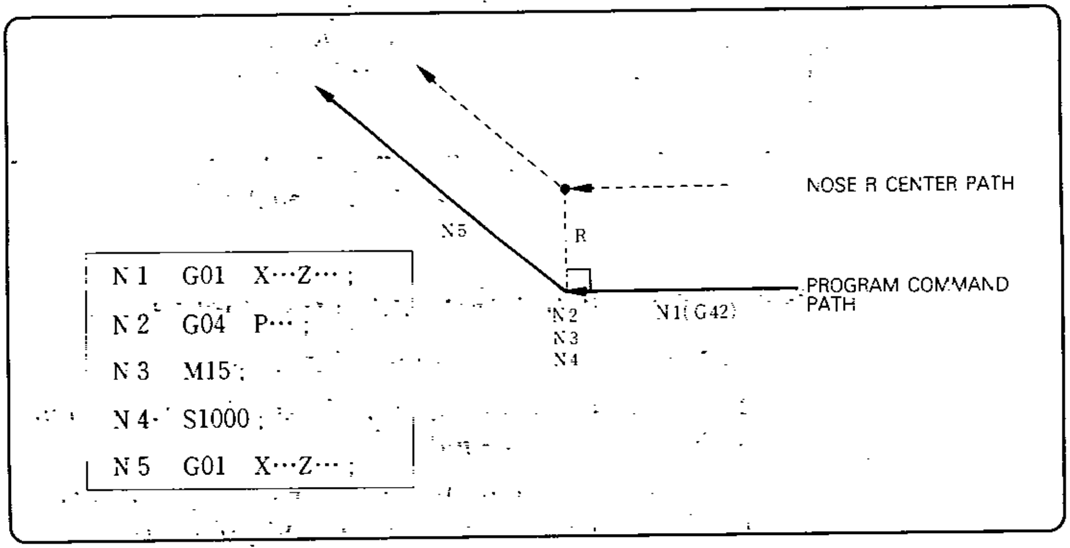


Fig. 2.12.39 Temporary Cancel Command during Nose R Offset (1)

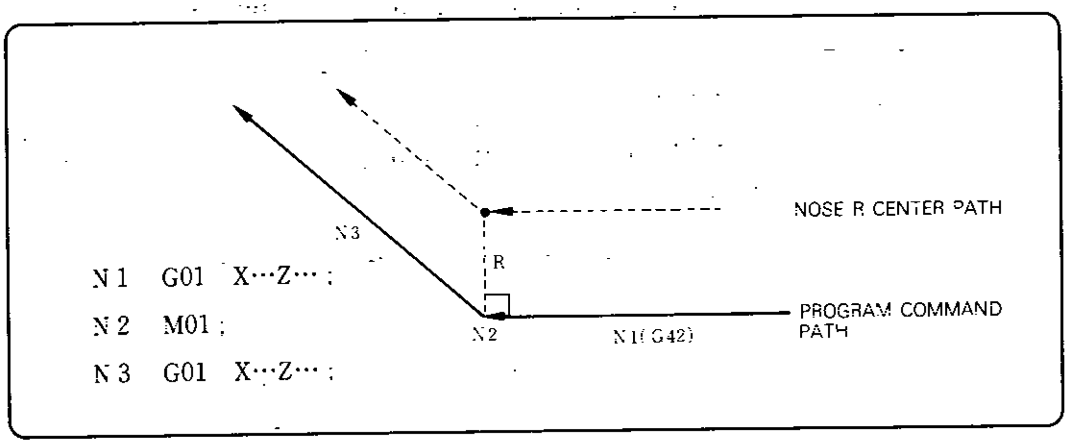


Fig. 2.12.40 Temporary Cancel Command During Nose R Offset (2)

(ii) Temporary cancel command (type II)

By the following commands, the nose R center moves so that the hypothetical tool nose point comes to the end point of the program command.

- Automatic reference point return command (G28, G30)
- Thread cutting command (G32, G34, G92)
- Coordinate system setting command (G50, etc.)

NOTE There is no move in the coordinate system setting command.

(iii) G41, G42 re-command

Commanding G41, G42 in the offset mode causes the nose R center to move to the R-offset position on the normal of the start point of the block immediately after, regardless of the following inside corner, outside corner, M96/M97 commands.

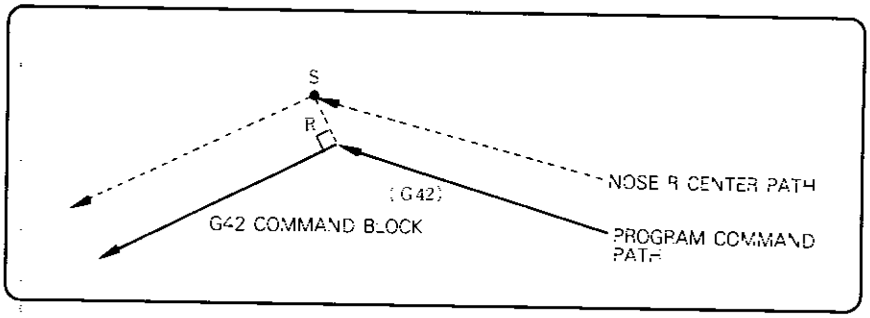


Fig. 2.12.41 G41, G42 Re-command During Nose R Offset

(iv) Command in offset mode without move

① The NC unit normally looks ahead of 2 blocks of data during nose R offset, to operate the tool path. If a block without coordinate command, as the G04 (dwell) block, is found, it looks ahead for one more block for operation. Up to 2 such blocks without coordinate command can be processed in this way.

In case coordinate command is not given for 3 blocks or more, the nose R offset operation cannot be executed, and the correct path cannot be obtained. When commanding G41, (G42), be careful not to let the move command in the offset plane discontinue for 3 or more blocks.

Example of Programming

```
      T··· ;
      G01  G41  Z···X···F··· ;
      Z···X··· ;
        :
      Z···X··· ;
G04   P1000 ;            ── Block without move in the
      Z···X··· ;            offset plane
        :                   (There can be two such blocks
      Z···X··· ;             causing trouble in the program)
      M··· ;  |
      S··· ;  |  ──
      Z···X··· ;
        :
      Z···X··· ;
      G40  Z···X··· ;
```

Fig. 2.12.42 Block without Move during Nose R Compensation

If there is no move command for 3 blocks or more, positioning is made for the offset amount on the normal of the end point in the block immediately before.

② When there are no move commands for 3 consecutive blocks, positioning is performed on the normal of the end point of the block immediately before. In case move command cannot be given on the offset plane for 3 consecutive blocks or more, and positioning on the normal is inconvenient, a dummy block can be inserted using I and K.

Example of Programming

N001 G17 G01 G41 Z······X······F······T······;
N002 Z······X······;
⋮
N010 Z······X······;
N011 K······I······; ······Dummy block
N012 M······; Command without
⋮ move (3 blocks or
N019 S······; more)
N20 Z······X······;
⋮
N29 Z······X······;
N30 G40 Z······X······;

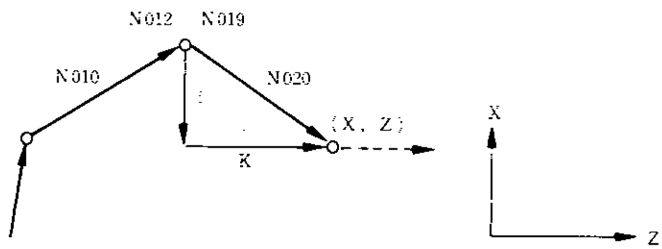


Fig. 2.12.43 Inserting Dummy Blocks

The dummy block is a block that includes no actual move and only gives the data necessary for nose R compensation operation. In the above example, the same command as the first block (N020) of the ZX move restarted after move of the Z-axis, is commanded as dummy by the I, K commands.

Addresses I, K are used for this dummy command, corresponding respectively to the X, Z axes.

I : X-axis command dummy
K: Z-axis command dummy
} Commanded by incremented value

NOTE I is commanded by radius value

In the above example, if N020 X.....Z..... is absolute command, change the commanded value to equivalent increment value.

NOTE When the dummy block includes circular interpolation, create the dummy block as shown in the following **Fig. 2.12.44**, and **2.12.45**.

(Example)

N050 G01 Z.......X....... ;

| N051 G01 K(b) I(—a) ; Dummy block |

N052 M....... ;

⋮ Command without move

N059 S....... ;

| N060 G03 Z.......X.......K(a) I(b) ; Circular interpolation |

N061 G01 Z.......X....... ;

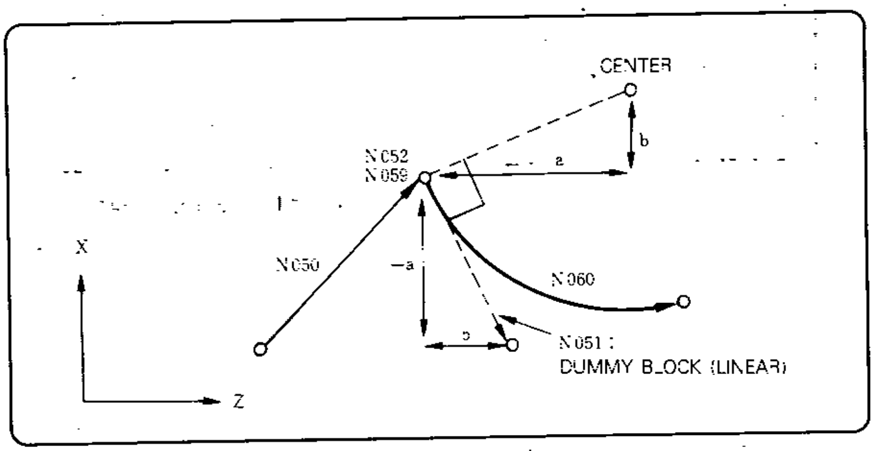


Fig. 2.12.44 Inserting the Dummy Block

As shown in **Fig. 2.12.43**, insert a linear dummy block that specifies the tangential direction at the start point of the commanded arc. Be very careful of the sign of the dummy block data. In some dummy blocks, it stops once at point A to prepare for the next arc designation (**Fig. 2.12.45**).

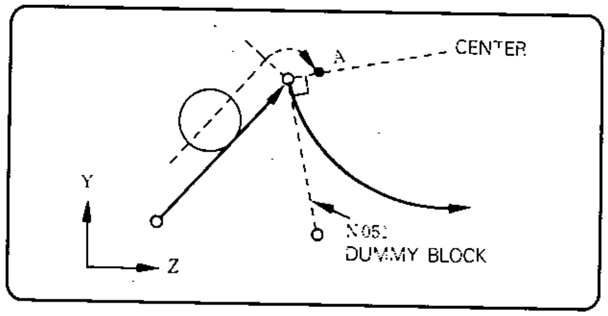


Fig. 2.12.45 Stops at Point A in Some Dummy Block

NOTES WHEN INSERTING DUMMY BLOCK

When dummy block is commanded by move command plus I, K under the offset mode, I, K are disregarded. However, when I, K are commanded when the offset is canceled, the offset position is corrected from point 1 to point 2 in I, K direction.

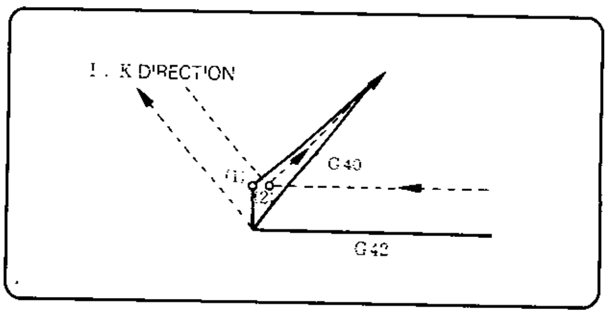


Fig. 2.12.46 Correcting the Offset Position

(v) Switching between G41, G42 in the offset mode

Switching between G41 and G42 can be performed directly in the offset mode, without canceling the offset by G40.

The following 2 types are offered, switched by parameter.

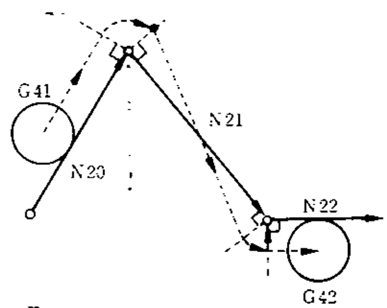
pm4013 D1 = 0 : type A
pm4013 D1 = 1 : type B

① **Type A : Switching is performed at the start and end of the direction switching block**

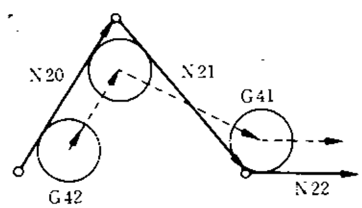
Example of Programming

```
N10  T··· ;
N11  G41 (G42) Z···X··· ;
          ⋮
N20  G01  Z···X···F··· ;
N21  G42 (G41) Z···X··· ;   ← Switching block
N22       Z··· ;
```

(a) G41 → G42 (M96 mode)



(b) G42 → G41



NOTE The same movement as the above can be obtained also by dividing the N21 block into two blocks as:

```
G42 (G41) ;
  Z······X······ ;
```

Fig. 2.12.47 Switching at the start and End of the Direction Switching Block

② Type B : Switching is performed at the intersection of the tool center path (offset path)

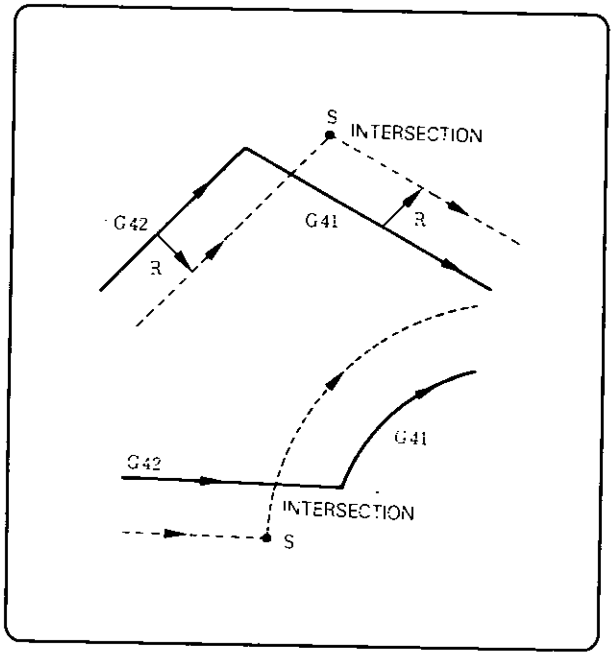


Fig. 2.12.48 Switching at the Intersection of the Tool Center Path

NOTES ON G41, G42

When there is no intersection, type A switching method is applied.

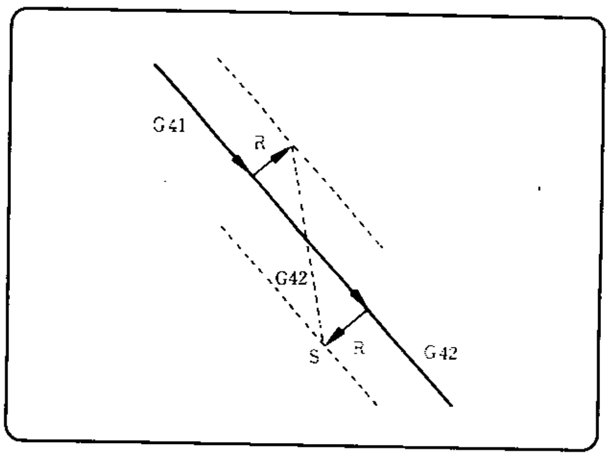


Fig. 2.12.49 Switching when there is No Intersection

(vi) Changing the tool offset amount in the offset mode

The tool offset amount can be changed in the following two ways, each switched by parameter.

pm4013 D2 = 0 : type A
pm4013 D2 = 1 : type B

① Type A : The new offset amount is effective from the block with T code command and the next block offset operation

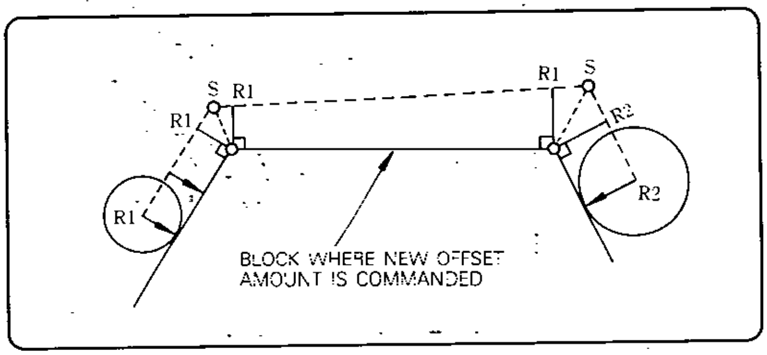


Fig. 2.12.50 The New Offset Amount is Effective from the Block with T Code Command and the Next Block Offset Operation

② Type B : The new offset amount is effective from the block prior to the T code command and the offset operation of the block with T code command

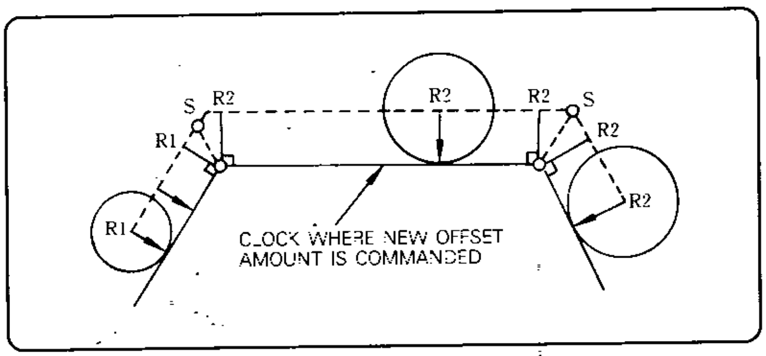


Fig. 2.12.51 The New Offset Amount is Effective from the Block Prior to the T Code Command and the Offset Operation of the Block with T Code Command

(7) Cancel

Cancel of the offset mode starts from the block with the T**00 or G40 command.

The block where the cancel mode starts is called the cancel block.

The following three types of cancel commands are offered.

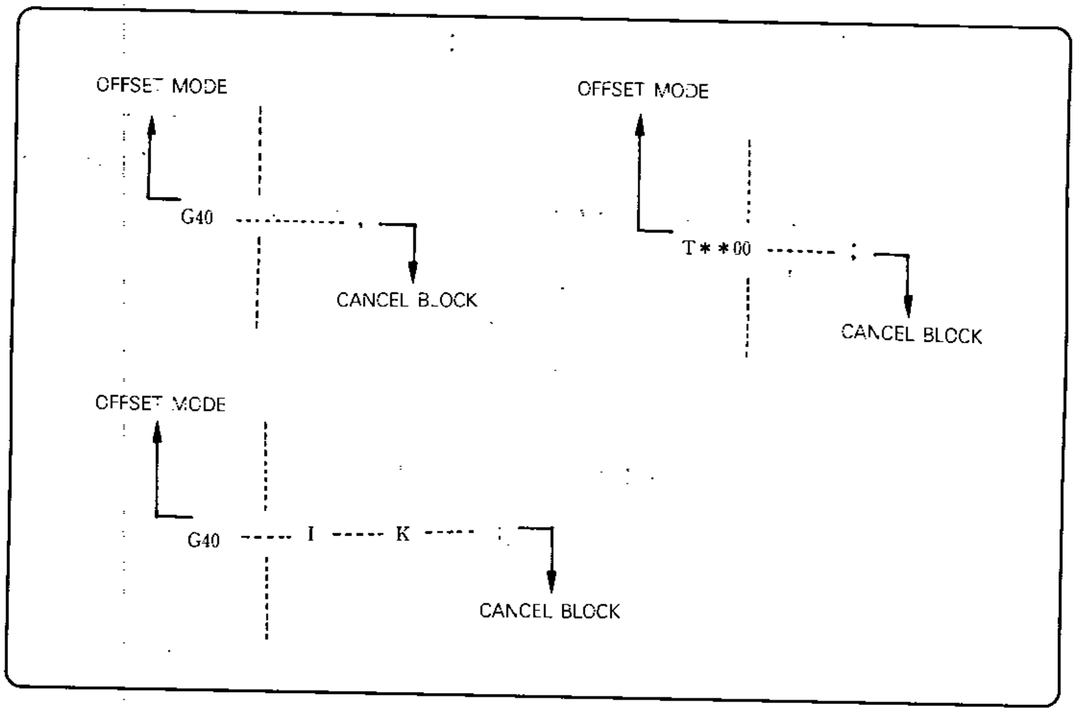


Fig. 2.12.52 Offset Cancel

Cancel movement

Two cases can be considered : when the block to be canceled has move command and when it does not

① When the cancel block has move command

(i) Inside corner (180 deg or less) cancel

The tool center moves onto the normal of the end point of the block immediately before the cancel block.

The command value and the hypothetical tool nose match in the cancel blocks.

```
 (G42)
G01 X....Z....
    X....Z....
G40 X....Z..... ;
```

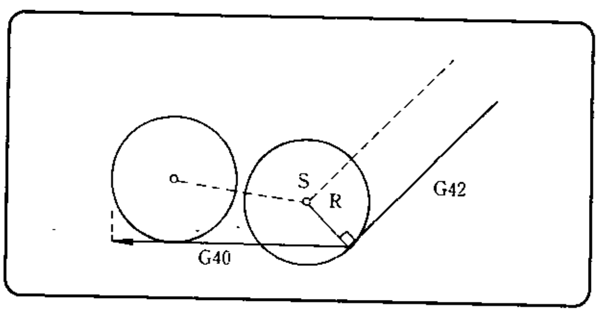


Fig. 2.12.53 Inside Corner (180 deg or less) Cancel (Linear)

```
Arc
(G42)
  :
G02 X...Z...R... ;
G40 X...Z... ;
```

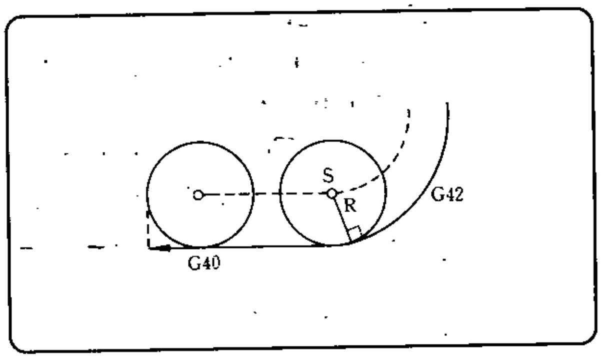


Fig. 2.12.54 Inside Corner (180 deg or less) Cancel (Arc)

(ii) Outside corner (180 deg or more) cancel

There are two types, selected by parameter.

```
pm4013 D0 = 1 : Type A
pm4013 D0 = 0 : Type B
```

This parameter is the same as the start up parameters.

[1] Type A : The tool nose center passes the offset position on the normal of the end point of the block immediately before the cancel block, and the hypothetical tool nose reaches the end point as programmed

(A-i)

```
(G41)
....................
G01  X......Z......F......;
G40  X .....Z ......;
```

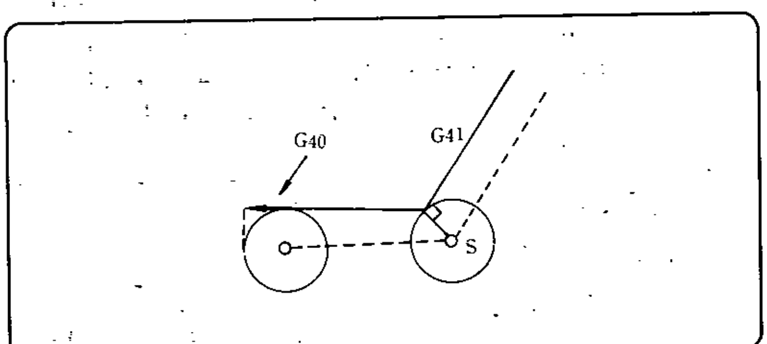


Fig. 2.12.55 Outside Corner (180 deg or more) Cancel 1

(A-ii)

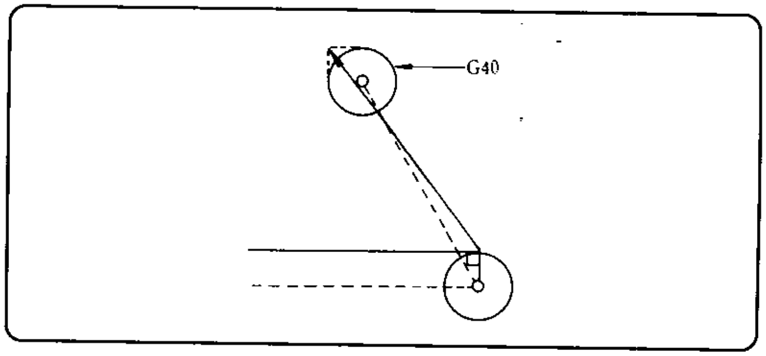


Fig. 2.12.56 Outside Corner (180 deg or more) Cancel 2

(A-iii)

```
G42
..................
G02  X.......Z.......I.......K....... ;
G01  G04  X........Z........ ;
```

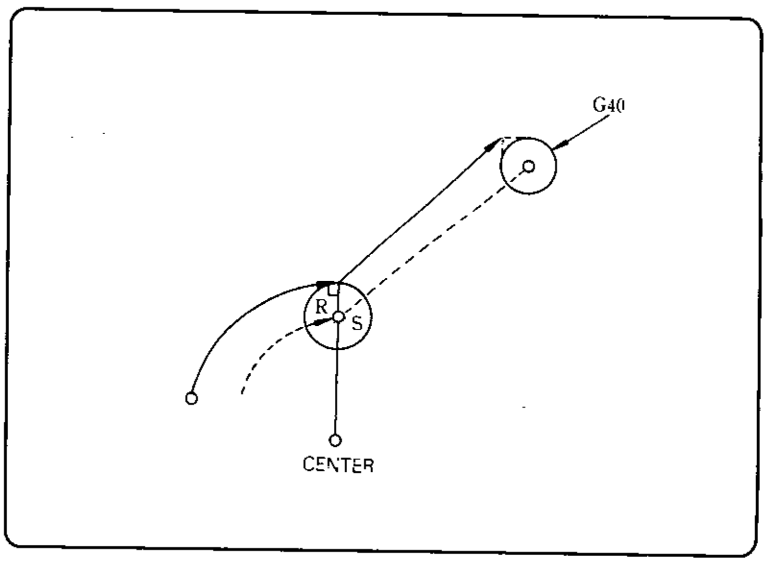


Fig. 2.12.57 Outside Corner (180 deg or more) Cancel 3

(A-iv)

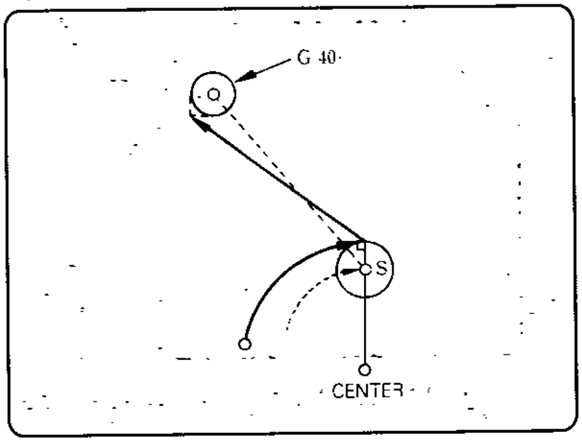


Fig. 2.12.58 Outside Corner (180 deg or more) Cancel 4

[2] Type B:

(B-i)

```
G42
.............
G01  Z.....F..... ;
G40  X.....Z..... ;
```

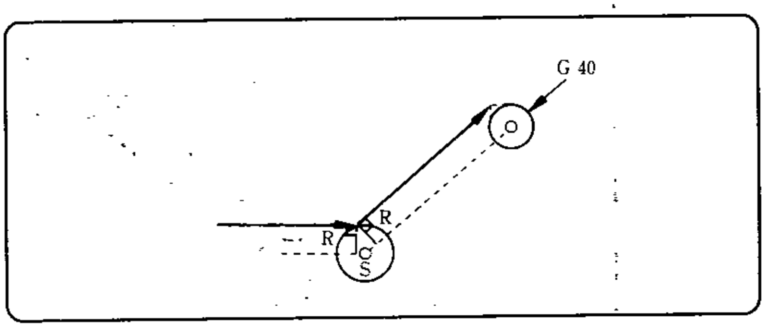


Fig. 2.12.59 Outside Corner (180 deg or more) Cancel 5

(B-ii)

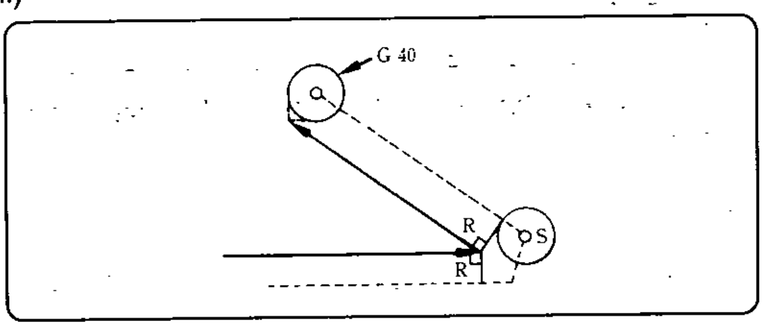


Fig. 2.12.60 Outside Corner (180 deg or more) Cancel 6

(B-iii)

```
G42
…………
G02  X……Z……I……K…… ;
G01  G40  X…..Z….. ;
```

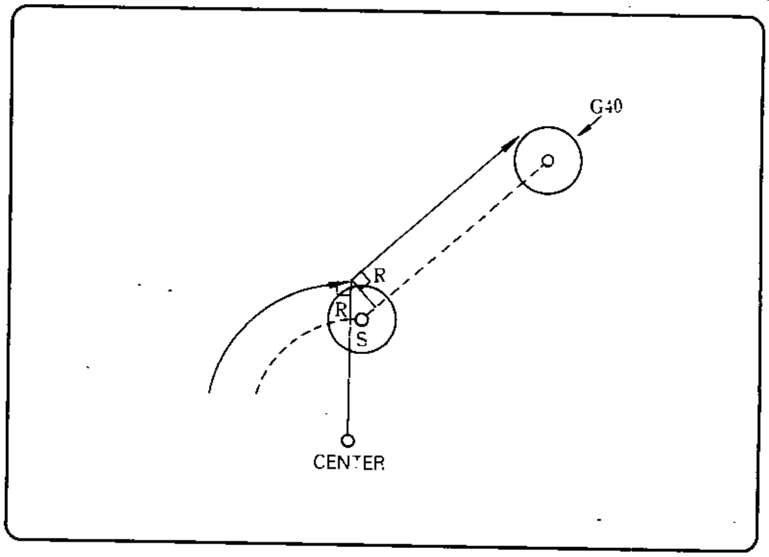


Fig. 2.12.61 Outside Corner (180 deg or more) Cancel 7

(B-iv)

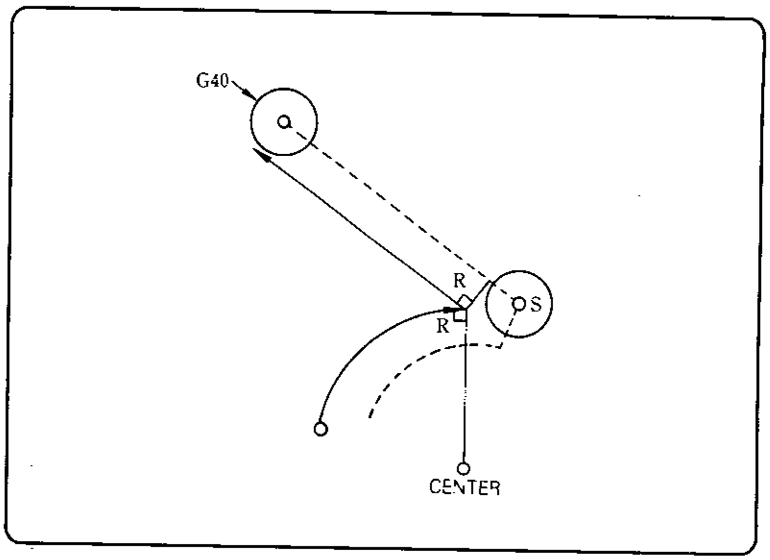


Fig. 2.12.62 Outside Corner (180 deg or more) Cancel 8

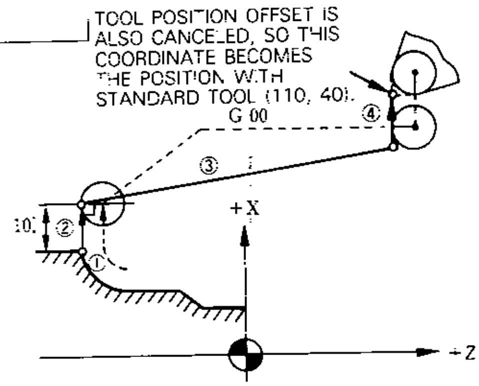
Example of Programming A

(G42, control point 3)

Offset mode

	:
①	G02
②	G01 U20. F0.25 ;
③	G00 G40 X110. Z40. ;
④	T0100 ;

← Movement at end of offset (with G00).

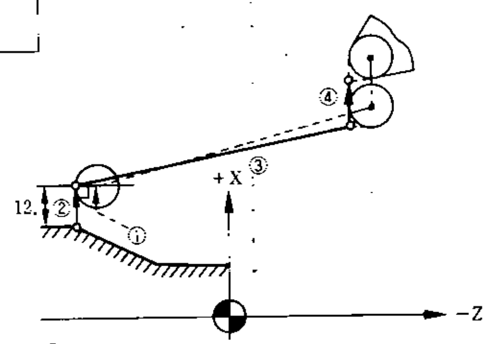


Example of Programming B

(G42, control point 3)

	:
①	G01 X······Z······F······ ;
②	G01 U24. F0.3 ;
③	G01 G40 X80. Z40. F6. ;
④	G00 T0200 ;

← Movement at end of offset (with G01)



② When the cancel block does not have move command

The hypothetical tool nose moves to the command point when ending the offset, also when the G40 block has no move command.

As the G40 (and T□□00) command includes move, always command G00 or G01 in the preceding or the same block. If the G code of group 01 is other than G00, G01 or G11, error "0181" occurs.

Example of Programming C

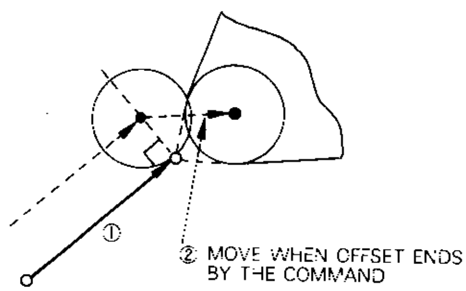
(G41, control point 4)

⋮

① G01 X ......Z ......F ...... ;

② G01 G40 F ...... ;

③ G00 T0300 ;



NOTE When canceling nose R compensation by the T□□00 command, tool position offset cancel starts as soon as the nose R compensation ends. The hypothetical tool nose moves to meet the final command position when the tool position offset was canceled. If the two moves are not to be mixed, cancel the move by G40.

③ Cancel by X...Z...I...K... ;

Giving the I, K command in the G40 block allows special cancel moves.

The intersection is calculated by the block immediately before G40 and the I, K vector block commanded by G40, so that the nose R center can pass the intersection in the cancel movement.

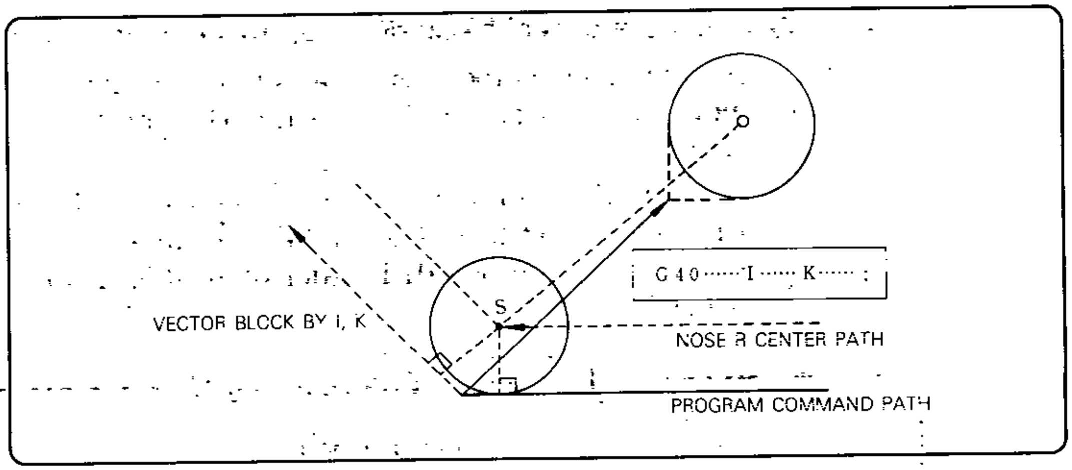


Fig. 2.12.63 Cancel by G40 I...K... Command

In this case, the nose R center always passes the intersection of the block immediately before G40 and the I, K vector block, regardless of the inside/outside corner state or the M96/M97 commands.

If the intersection cannot be obtained, the nose R center moves to the R-offset position on the normal of the end point of the block immediately before G40, as shown in **Fig. 2.12.64**.

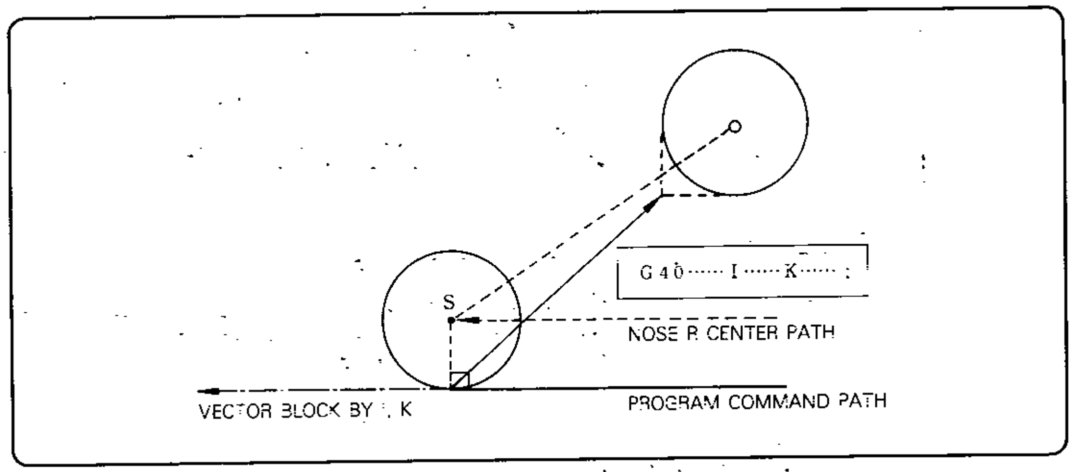


Fig. 2.12.64 When the Intersection Cannot be Found in Cancel by G40 I...K...Command

(8) Interference check

Interference check is offered to prevent the tool cutting into (interfering with) the material. When interference is suspected, alarm occurs (type A) or the tool center path is corrected (type B).

Types A and B are switched by parameter.

pm4013 D3 = 1	Type A
pm4013 D3 = 0	Type B

(a) Definition of interference

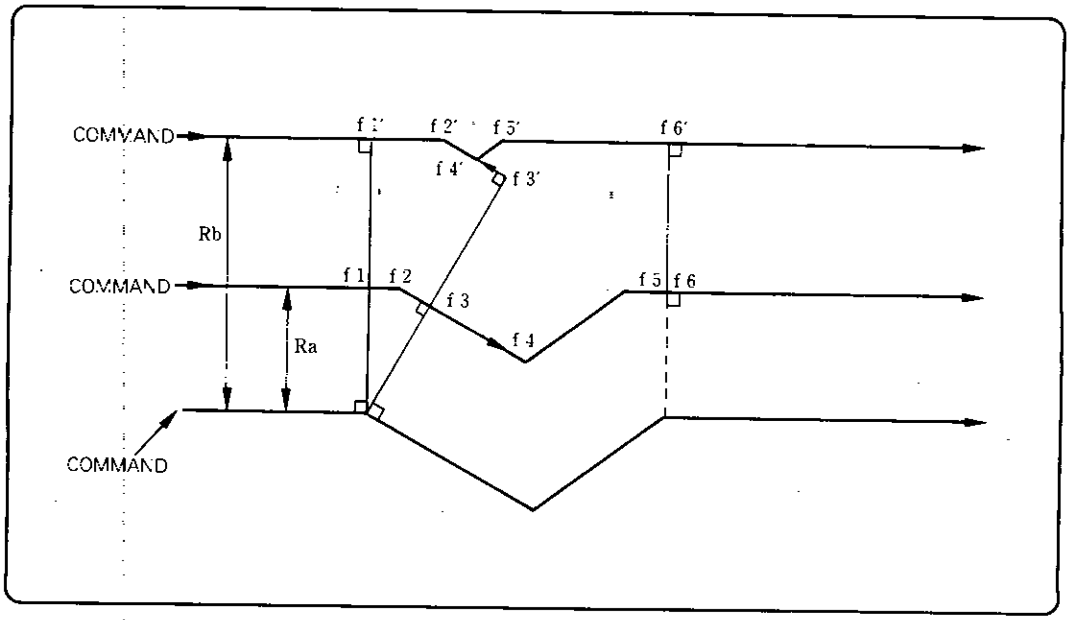


Fig. 2.12.65 Definition of Interference

The tool path when the nose R compensation amount is Ra is $f_1 \rightarrow F_2 \rightarrow F_3 \rightarrow F_4 \rightarrow F_5 \rightarrow F_6$ in the above command (**Fig. 2.12.65**), but when the nose R compensation is Rb, $f_3' \rightarrow f_4'$ differs 180 degrees from the correct command direction $f_3 \rightarrow f_4$ in the $f_1' \rightarrow f_6'$ path.

Such a case is regarded as interference, and alarm occurs.

(b) Type A : Alarm occurs

(i) Cutting in occurs at end point (2) of N1, and alarm "187" occurs. Then, it stops at the start point of N0 (1).

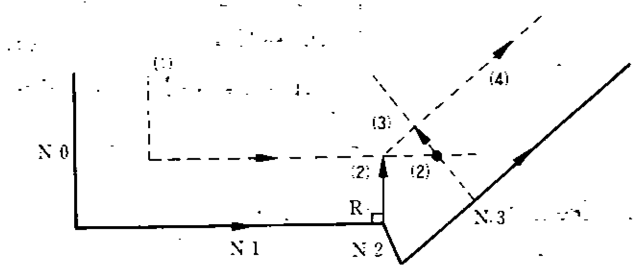


Fig. 2.12.66 Alarm Occurs 1

(ii) Cutting in occurs at the current N1 end point (2), and alarm "187" occurs. Then, it stops at N0 start point (1).

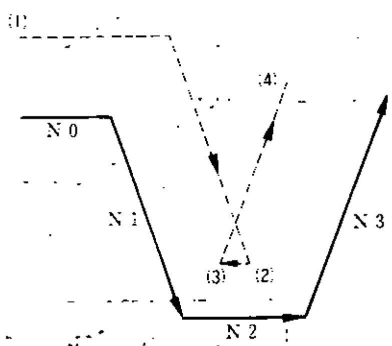


Fig. 2.12.67 Alarm Occurs 2

As shown in Fig. 2.12.66 and 2.12.67, cutting in (interference) is predicted when the move direction of the programmed block and the offset path differ too much.

NOTE No interference check is made upon start-up.

(c) Type B : Automatic correction when interference occurs

When interference is predicted after nose R compensation operation, the move that causes interference is erased and a path that does not cause interference is created.

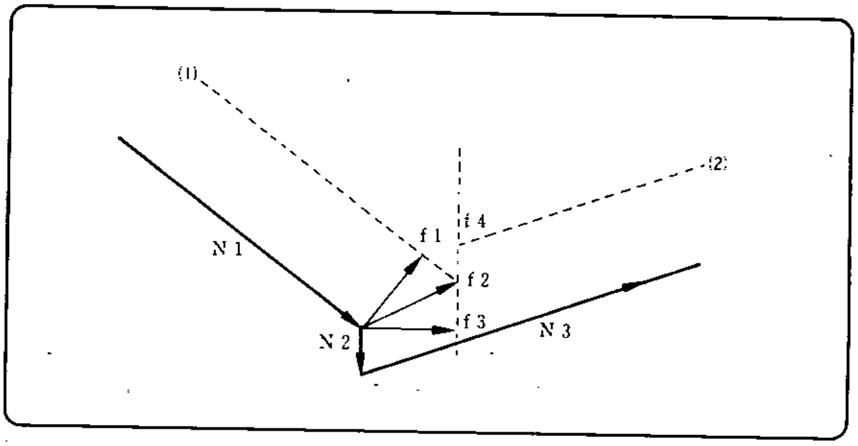


Fig. 2.12.68 Automatic Correction in Case Interference is Predicted

In **Fig. 2.12.68**, the three points, f_1, f_2, f_3 are created at the joints of N1 to N2 by nose R compensation operation.

At N2 to N3, f_4 is created. Then interference check is made on the four points f_1 to f_4. Points where interference occurs are erased in sequence, and the tool center path is created by connecting the remaining points.

Check at f_3 — f_4 ..... Interference — f_3 is erased
Check at f_2 — f_4 ..... Interference — f_2 is erased
Check at f_1 — f_4 ..... No interference occurs
It moves as (1) → f_1 → f_4 → (2)

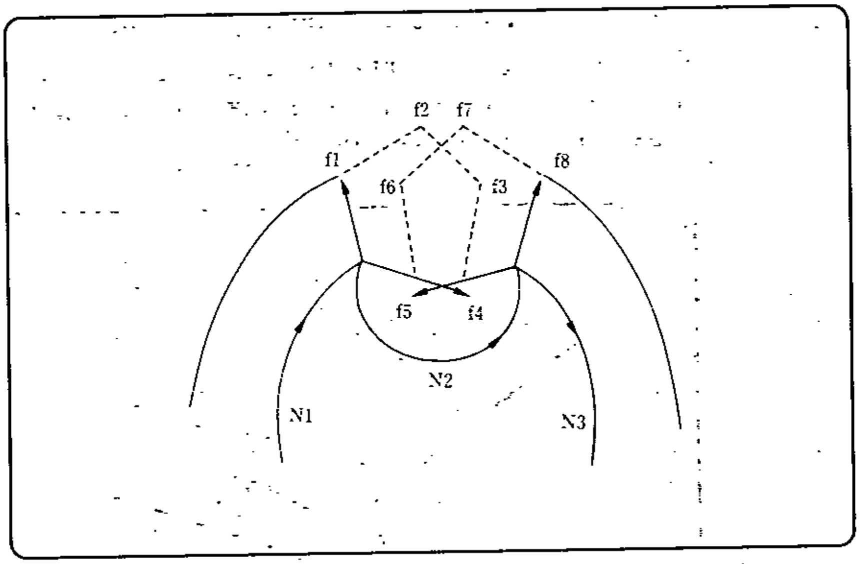


Fig. 2.12.69 Automatic Correction when Interference Occurs 2

At N1 — N2: f_1, f_2, f_3, f_4
At N2 — N3: f_5, f_6, f_7, f_8

Check at f_4 — f_5 ..... Interference — f_4, f_5 are erased Check at f_3 — f_6 ..... Interference — f_3, f_6 are erased Check at f_2 — f_7 ..... No interference occurs It moves as $f_1 \rightarrow f_2 \rightarrow f_7 \rightarrow f_8$

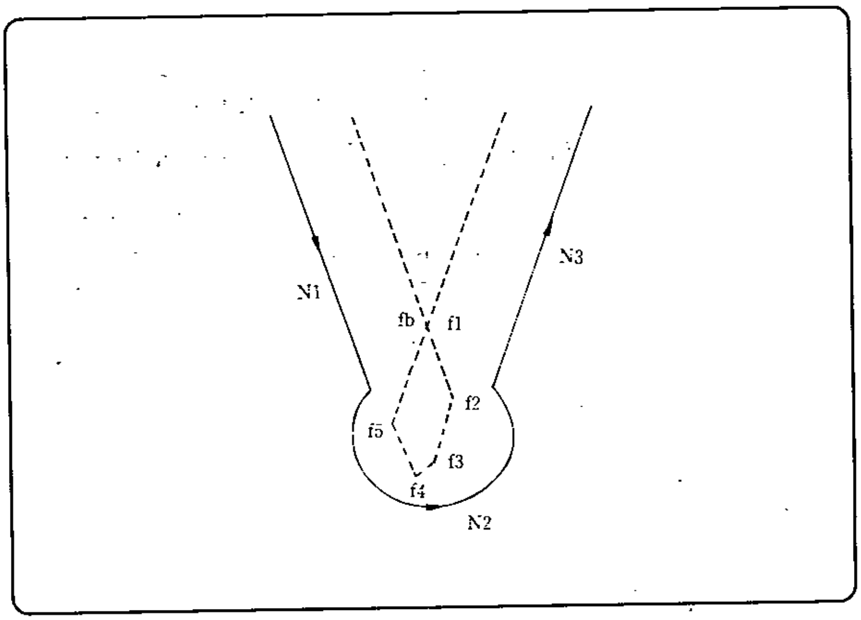


Fig. 2.12.70 Automatic Correction when Interference Occurs 3

At N1 — N2: f_1, f_2, f_3
At N2 — N3: f_4, f_5, f_6

Check at f_3 — f_4 ..... Interference — f_3, f_4 are erased
Check at f_2 — f_5 ..... Interference — f_3, f_5 are erased
Check at f_1 — f_6 ..... No interference occurs
It stops at the start point of block N1 where the alarm occurred.

When all the joint points are erased by interference check as in **Fig. 2.12.71**, alarm "188" occurs unconditionally to stop the operation.

NOTES ON INTERFERENCE CHECK

1. **Restrictions against the maximum move command value of the coordinate word**

   The restrictions against the maximum move command value (Table 2.1.12) of the coordinate word are the same as in nose R compensation.

2. **Command forms that cause alarm**

   The following command forms cause alarm "184".

   (a) When the following kind of arc is commanded in inside offset of arc command

   Command circle radius $r+5 \leq$ Nose R radius R

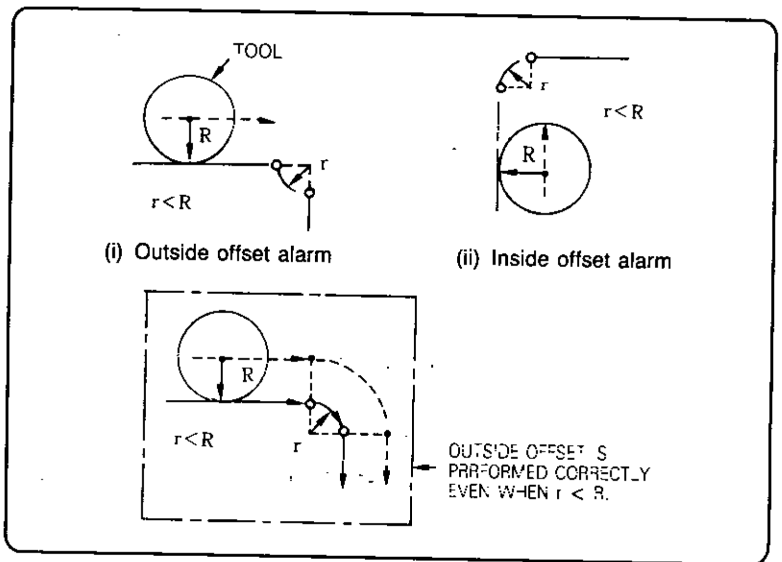


Fig. 2.12.71 Command form that Causes Alarm 1

NOTES ONINTERFERNCE CHECK (Cont'd)

(b) When there is no intersection on the offset tool center path

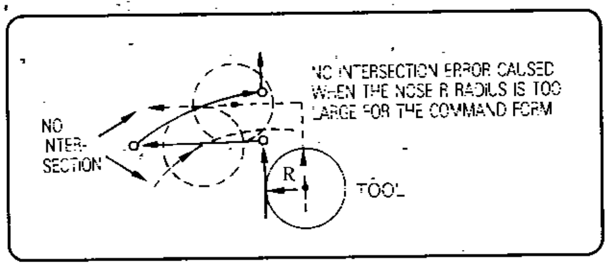


Fig. 2.12.72 Command form that Causes Alarm 2

3. **G codes that can be used during nose R compensation**

   Table 2.12.5 shows the G codes that can be used during nose R compensation. As a rule, do not specify G codes other than the following.

Table 2.12.5 G Codes that Can be Used during Nose R Compensation

G Codes that Can be Used	Remark
G00, G01, G04, G06, G11 G96, G97 .... Constant surface speed control G98, G99 .... Feed function designation G90, G91 .... Absolute, incremental command	
G02, G03 G12, G22, G23 .... Commands including circular interpolation G70, G71, G72, G73 .... Multiple repetitive cycles G111, G112 .... Complex chamfering, rounding	Cannot be used in the compensation start and cancel blocks

The following G codes cannot be commanded during nose R compensation. If so, alarm "0161" occurs.

- G31
- G74.G75/G76
- G68/G69
- G122/G123

4. **Circular interpolation by radius designation**

   Nose R compensation can also be commanded against circular interpolation by radius designation.

5. **Subprogram command during compensation mode**

   Subprograms (M98, M99) can also be commanded in the compensation mode.

NOTES ON INTERFERENCE CHECK (Cont'd)

6. **Nose R compensation memory**

   The number of nose R compensation amount memory differs in the basic and optional configuration. The maximum nose R compensation amount is ±99.999 mm (or 9.9999 in).

7. **Offset of difference smaller than nose R compensation under the M96 mode**

   If offset of difference smaller than nose R compensation is made under the M96 mode, excess infeed occurs. Cutting under the M97 mode leaves some parts uncut, though the M97 mode seems to be more practical for machining.

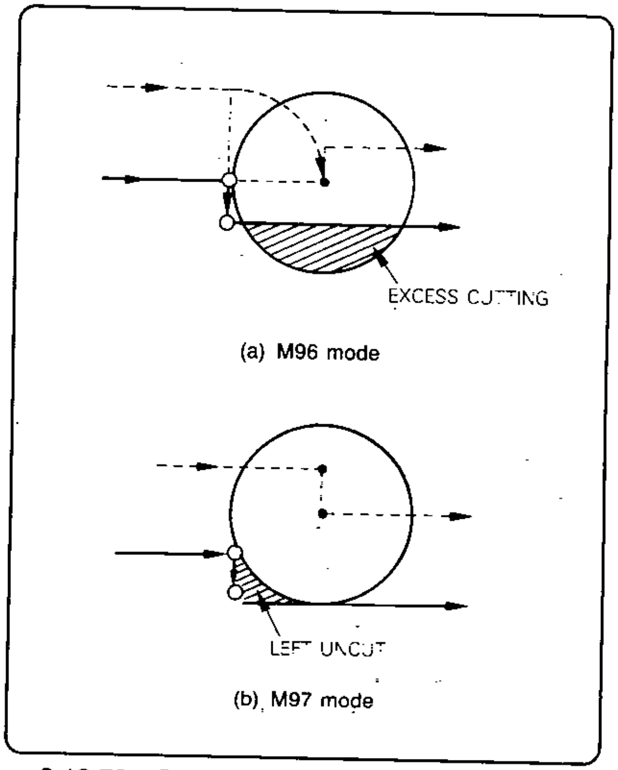


Fig. 2.12.73 Compensating Difference Smaller than the Nose R Amount

8. **M96 mode**

   Even when under the M96 mode, if both the ΔX and ΔZ are smaller by a certain amount as in **Fig. 2.12.74**, no corner rounding it performed, and the tool moves to a point on line B. The certain amount is preset in parameter #4450.

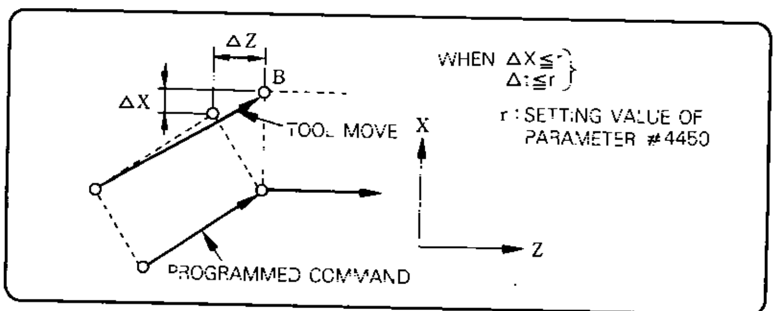


Fig. 2.12.74 Tool Path when Both ΔX and ΔZ are Smaller than a Certain Value

NOTES ON INTERFERENCE CHECK (Cont'd)

9. **Interruption of MDI operation during nose R compensation mode**

   No interruption can be made by MDI operation during the nose R compensation mode.

10. **Interruption to the actual buffer under the nose R compensation mode**

    When under the G41 or G42 offset mode, the single-block switch can be turned on to stop the block, and the RAPID or JOG mode called, to write the following data using the same procedure as the MDI writing procedure.

Data that can be written:	F, M, S, T codes
Block to be written into:	Added to the block with the currently executed actual buffer

After writing, depress the cycle start button still under the RAPID or JOG mode. The written code is executed immediately to send signals as the BIN codes. When returning to the original automatic operation mode, start the cycle, to continue the automatic operation.

NOTE The following M codes cannot be written.
M00, M01, M02, M30, M codes processed inside the NC

T code command which tool offset number is "00"

T□□00;

The commands have the following meanings.

① Cancel tool position offset.

② Cancel nose R compensation.

Program the following command.

```
N 2   G 41 ;
N 3   G 00  T 0101 ; ┐
        :            │ UNDER NOSE R COMPENSATION
        :            │ COMMAND FOR TOOL NO. "01"
N 21  G 00  T 0100 ; ┤
                     │ CANCELING NOSE R COMPENSATION,
                     │ TOOL POSITION OFFSET
N 25  G 00  T 0202 ; ┤
        :            │ UNDER NOSE R COMPENSATION
        :            │ COMMAND FOR TOOL NO. " 02"
N 40  G 00  T 0200 ; ┘
N 41  G 40 ;
```

Fig. 2.12.75 Cancel by T□□00

NOTES ON INTERFERNCE CHECK (Cont'd)

```
N 1   G50   X140.  Z20. ;
N 2   G00   S1700  M03  T0202 ;
N 3   (G00) G42  X 0   Z 5 . ; ←Nose R compensation start block
N 4   G01   Z 0   F0.2 ;
N 5   X20. ;
N 6   Z−20. ;
N 7   X30.  W−15.  S1100 ;
N 8   G12   W−20.  13. ; ← (Rounding)
N 9   G11   X50.  K− 3 .  S700 ; ← (Chamfering)
N10   G01   Z−70. ;
N11   G02   X90.  Z−90.  R20.  S360 ; ←(R designation on arc)
N12   G01   X110.  S300 ;
N13   G04   U 0 ; ← (Dwell: Angling)
N14   (G01) Z−110. ;
N15   X120. ;
N16   G00   X140.  Z30.  T0200 ; ←Nose R compensation end block
N17   G40 ;
```

Under nose R compensation mode (N4–N15)

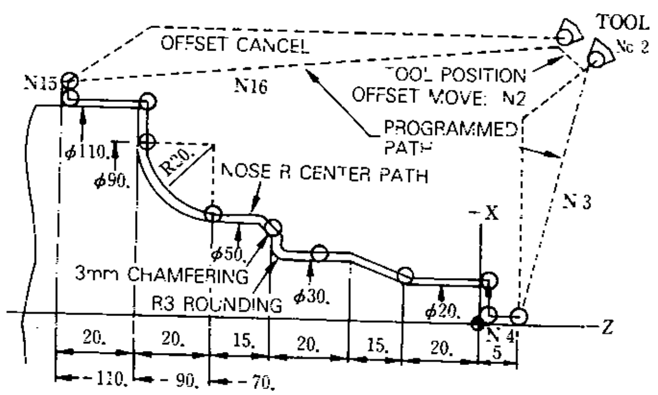


Fig. 2.12.76 Program Example

2.12.4 Multi-active Register (M92, M93) *

Uses the M codes as shown in **Table 2.12.6**.

Table 2.12.6 Multi-active Registers

M code	Description
M92	Multi-active register off
M93	Multi-active register on

M93 ;

When M93 is commanded, it looks ahead for 7 blocks until M92 is commanded.

This can avoid stop between blocks when the operation time of the 7-block look ahead is longer than the look-ahead operation time in certain programs.

M92 ;

Cancels the 7-block look-ahead mode.

2.12.5 Corner Rounding Judgment Internal M Code (M96/M97)

Table 2.12.7 shows the corner rounding judgment internal M codes.

Table 2.12.7 Corner rounding judgment internal M code

M code	Description
M96	Nose R compensation corner rounding ON
M97	Nose R compensation corner rounding OFF (Execution of intersection computation)

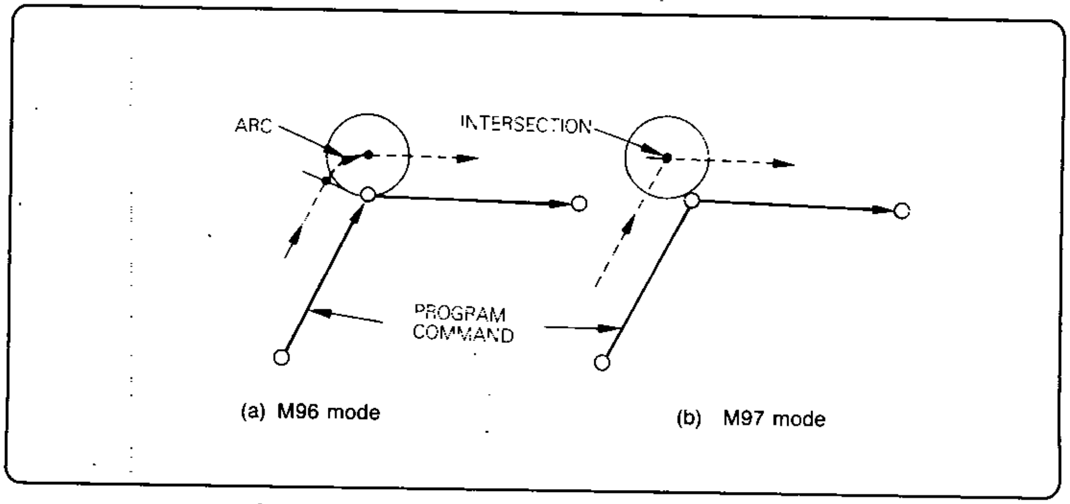


Fig. 2.12.77 Movement in M96, M97 Modes

(1) Movement during nose R compensation mode by G41 or G42

When a form has a corner in which the tangential angle is 180 degrees or more, the corner is rounded in the M96 mode of nose R compensation mode by G41 or G42.

The tool path is not an arc. It operates an intersection shifted for the nose R amount, and offsets the move so that it passes the intersection.

(2) M96, M97 commands are modal

The M96, M97 commnds are modal. The fault occurs when the power is initially supplied is M96.

(3) Block when M96, M97 become effective

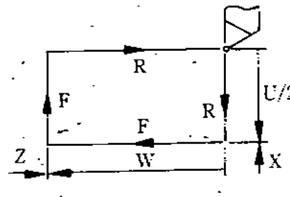
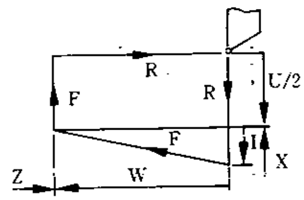
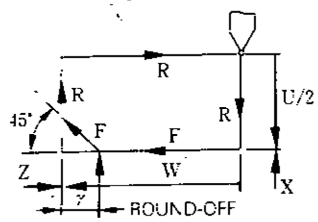
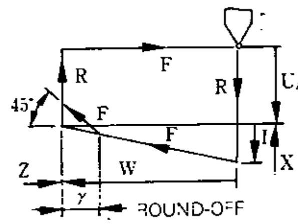
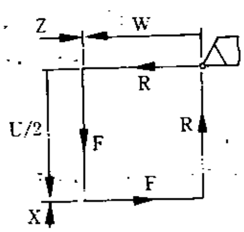
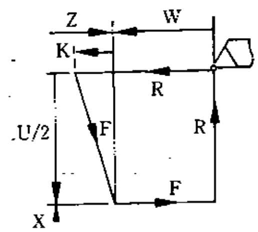
The move command block where M96, M97 become effective is as follows.

G01 Z......X......F...... ; (G01) Z......X......M96 ;	M96 is effective from the corner move of these 2 blocks
Z.....X. ..... ; Z .....X .....M97 ;	M97 is effective from the corner move of these 2 blocks

2.13.1 Canned Cycle (G90, G92, G94)

Basic cutting: By canned cycle, operation equivalent to the four blocks of "infeed → cutting (or thread cutting) → clearance → return ...." is performed as 1 cycle by single-block command. G90, G92, G94 are offered (**Table 2.13.1**).

Table 2.13.1 Cycle

G code	Straight Cycle	Taper Cycle
G90 Outer Diameter Cutting	G90 X(U)........Z(W)........F(E)......... ; 	G90 X(U) ......Z(W) ......I ......F(E) ...... ; 
G92 Thread Cutting	G92 X(U).........Z(W).........F(E).........; 	GG92 X(U)......Z(W)......I......F(E)...... ; 
G94 Front Cutting	G94 X(U)........Z(W)........F(E)........ ; 	GG94 X(U).....Z(W).....I.....F(E)...... ; 

2.13.1.1 Cutting cycle A (G90)

(1) Straight cutting cycle

G90 X(U)…….Z(W)…….F(E)……. ;

The above command executes the cycle of 1 to 4 in **Fig. 2.13.1**.

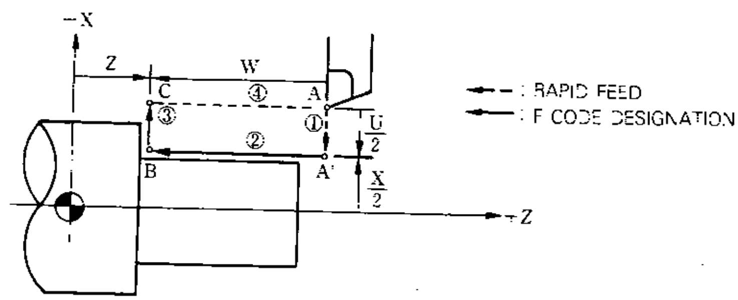


Fig. 2.13.1 Straight cutting cycle

As the G90 command is modal, the cycle is executed from the next block on, just be commanding the X-axis direction infeed as:

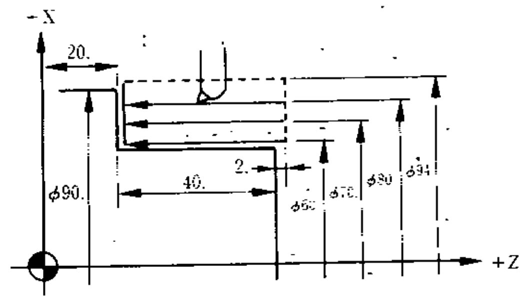
X(U)………. ;
X(U)………. ;

Example of Programming

```
N10  G00  X94.  Z62. ;

N11  G90  X80.  W-42.  F0.3 ;  ←— Start of G90 cycle
N12  X70. ;                       } Change the cut-in and
N13  X60. ;                       } execute G90

N14  G00············;
```



(2) Taper cutting cycle

| G90 X(U).......Z(W).......I.......F(E)....... ; |

The above command executes the cycle of 1 to 4 in **Fig. 2.13.2.**

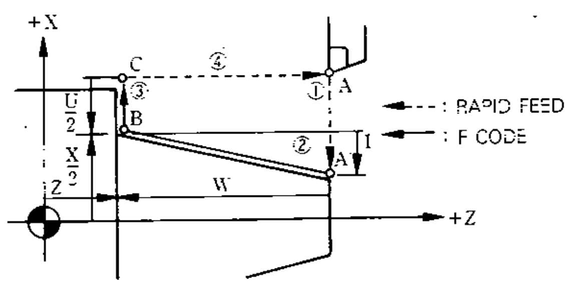


Fig. 2.13.2 Taper Cutting Cycle

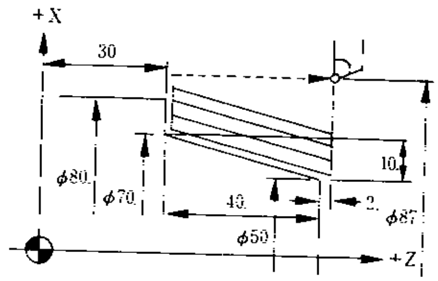
The sign of I is to be specified as viewed from point A' with point B as reference.

Example of Programming

```
N20  G00  X87.  Z72. ;

N21  G90  X85.  W-42.  I-10.5  F0.25 ;
N22  X80. ;
N23  X75. ;
N24  X70. ;

N25  G00············ ;
```



NOTES ON G90

1. **Designation of the T, S, M functions**

   The T, S, M functions whuch will act as the cutting conditions of the G90 cycle are basically to be specified before the G90 block.

   However, T, S, M blocks without move command in the area which G90 is effective are effective.

```
G90 X······Z······I······F(E)······ ;
    X······ ;
    X······ ;                                   G90 is effective
    X······T0505 M05 ; ← Error
G00 X······Z······ ;
```

```
G90 X······Z······I······F(E)······ ;
    X······ ;
    X······ ;                                   G90 is effective
G00 X······T0505 M05 ; ← Correct
    X······Z······ ;
```

   The G90 effective area starts from the where G90 is specified and ends at the block immediately before the block where another group 01 G code is specified.

   This also applies to the following G92 and G94.

2. **G90 cycle when single-block switch is "ON"**

   When single-block switch is "ON", the G90 cycle finishes the 1 to 4 cycle without interruption, and stops after completing the cycle.

2.13.1.2 Thread cutting cycle (G92)

(1) Straight thread cutting cycle

G92 X(U)……Z(W)……F(E)…… ;

F(E) — Specify the thread lead (L)

The thread cutting cycle of the following **Fig. 2.13.3** (1 to 4) is executed by the above command.

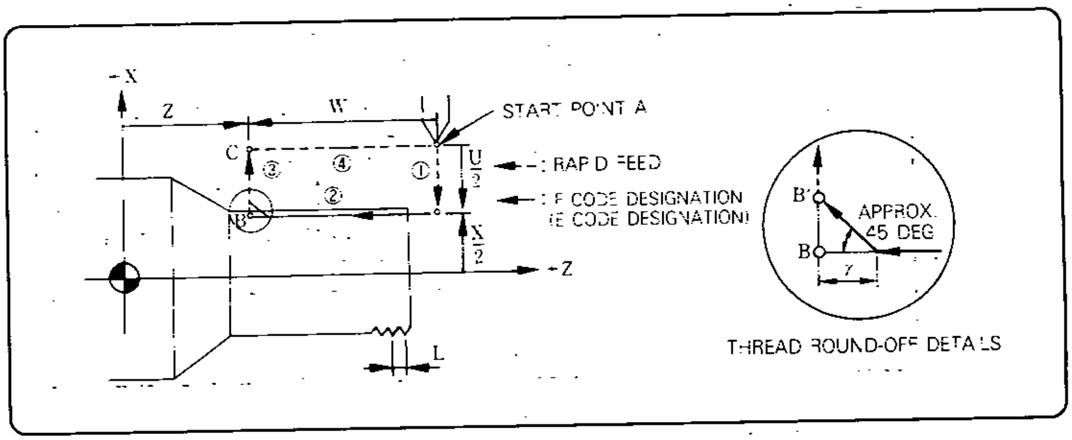


Fig. 2.13.3 Straight Thread Cutting Cycle

⟨Thread round-off⟩

The thread is rounded off if thread round-off input (CDZ) is on when the G92 command is given.

The thread rounding off amount γ can be set by parameter #0100 in 0.1L unit within the range of 0 to 25.5L.

L is the lead of the commanded thread.

It is generally recommended to program a sequence using optional M code to turn on/off the thread round-off input (CDZ).

As G92 is modal, the thread cutting cycle is executed in the following blocks just by commanding infeed in the X-axis direction by;

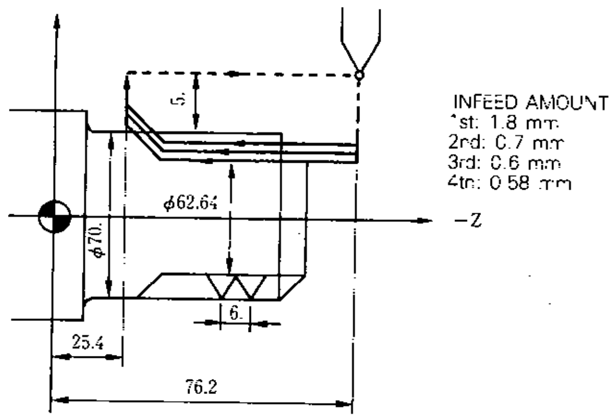
X(U)…… ;
X(U)…… ;

Example of Programming

N30 G00 X80. Z76.2 M○○; —— M○○; Thread round-off be on.

N31 G92 X66.4 Z25.4 F6. ;	
N32 X65. ;	
N33 X63.8 ;	Thread cutting cycle
N34 X62.64 ;	4 infeeds

N35 G00 X100. Z100. M△△ ; ←— M△△ ; Thread round-off must be off.



(2) Straight thread cutting cycle (thread angle infeed)

G92 X(U).......Z(W).......K.......F(E)....... ;

The above command starts thread cutting according to the thread angle. The cycle (1 to 4) in **Fig. 2.13.4** is executed.

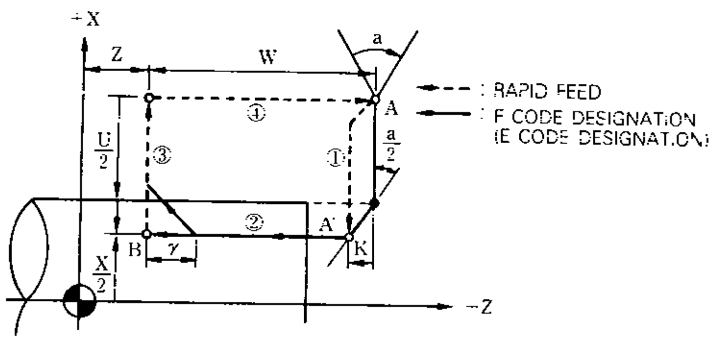


Fig. 2.13.4 Straight Thread Cutting Cycle (Thread Angle Cutting)

The Z-axis direction shift amount K of point A and point A' can be specified by a value with sign with point A as reference. (Fig. 2.13.5)

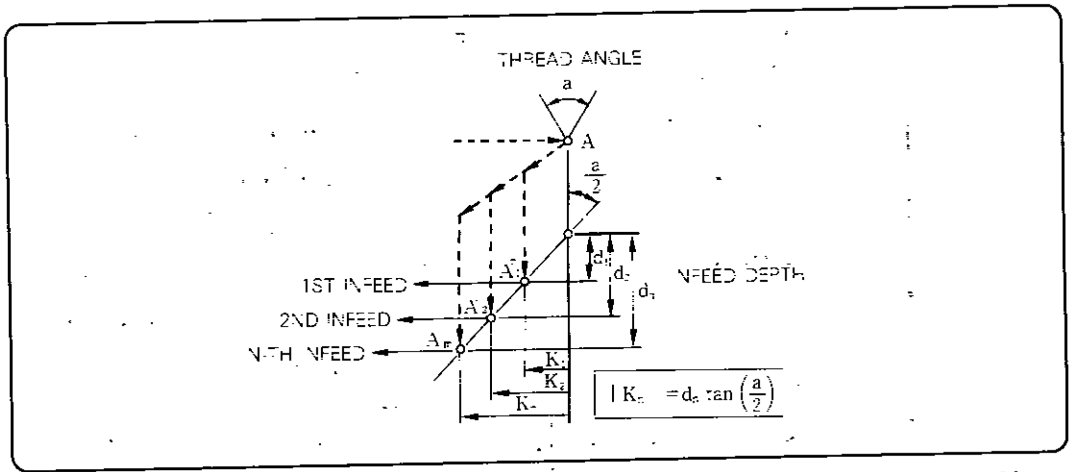


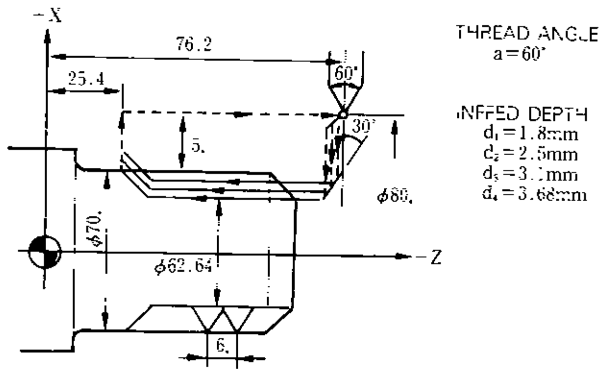
Fig. 2.13.5 Specifying the Z-axis Shift Amount K of Point A and Point A'

Calculate and program the value of K for each infeed as shown in Fig. 2.13.4 so that the thread cutting can be performed along the thread angle (a).

Table 2.13.2 Cross Reference for Thread Angle (a) and Tan (a/2)

A	Tan (a/2)
29°	0.258618
30°	0.267949
55°	0.520567
60°	0.577350
80°	0.839100

Example of Programming



Calculating K .....$|K| = d \tan(60°/2)$

$K_1 = -1.8 \times 0.57735 = -0.866$mm

$K_2 = -2.5 \times 0.57735 = -1.443$mm

$K_3 = -3.1 \times 0.57735 = -1.790$mm

$K_4 = -3.68 \times 0.57735 = -2.125$mm

Therefore, the following program is to be written.

```
N40  G00  X80.  Z76.2  M○○ ;

N41  G92  X66.4  Z25.4  K-0.87  F6. ;
N42  X65.  K-1.44 ;
N43  X63.8 K-1.79 ;
N44  X62.64 K-2.13 ;

N45  G00  X100.  Z100.  M△△ ;
```

(3) Taper thread cutting cycle

| G92 X(U).......Z(W).......I.......F(E)....... ; |

The above command executes the taper thread cutting cycle (1 to 4) as shown in **Fig. 2.13.6.**

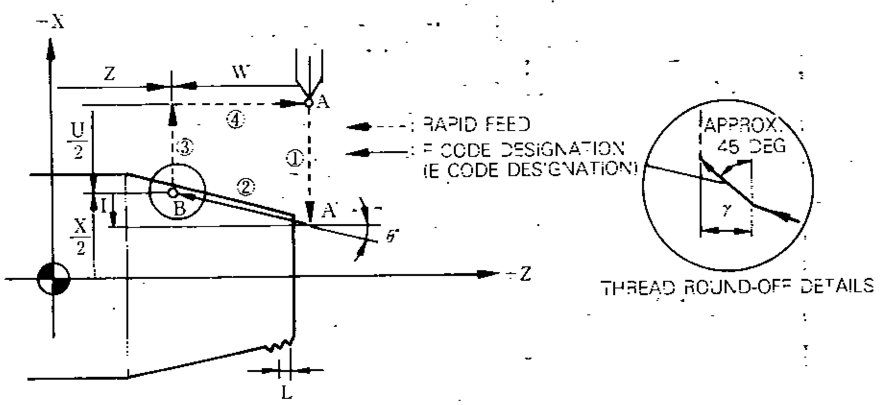


Fig. 2.13.6 Taper Thread Cutting Cycle

As G92 is modal, the thread cutting cycle is executed in the following blocks just by commanding infeed in the X-axis direction by;

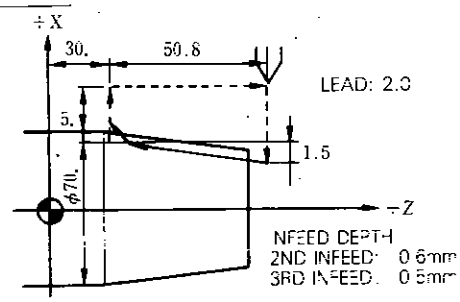
| X(U)...... ; |
| X(U)...... ; |

Example of Programming

N50 G00 X80. Z80.8 M○○ ;

N51 G92 X70. W-50.8 I-1.5 F2. ;
N52 X68.8 ;
N53 X67.8 ;

N54 G00 X100. Z100. M△△ ;



(4) Taper thread cutting cycle (thread angle infeed)

G92 X(U)…….Z(W)…….I…….K…….F(E)……. ;

The above command specifies taper thread cutting according to the thread angle, as shown in **Fig. 2.13.7.**

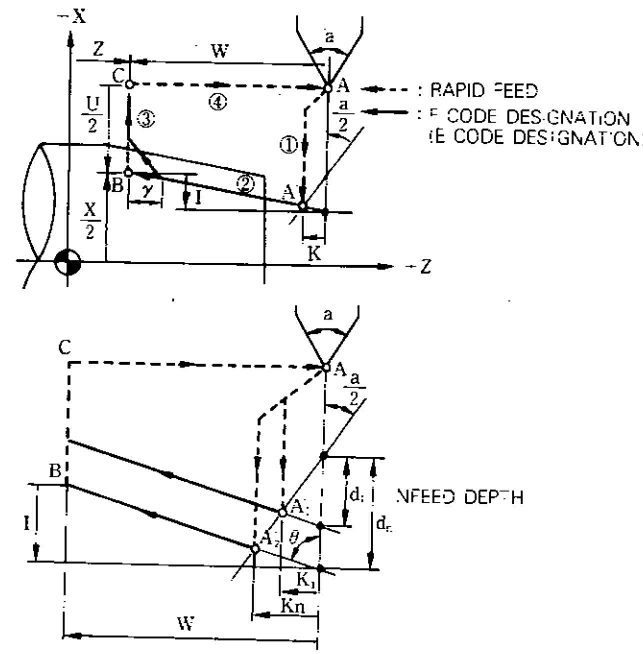


Fig. 2.13.7 Taper Thread Cutting Cycle (Thread Angle Cutting)

The K value must be calculated and programmed for each infeed, so that the infeed is made correctly according to the thread angle (a).

$$|Kn| = \frac{dn \tan(a/2)}{1 \pm \left|\frac{I}{W}\right| \bullet \tan(a/2)}$$

Note that the sign of the divider (θ') is determined as follows.

"+" when θ' < 90 deg
"−" when θ' > 90 deg

NOTE Calculation will be rather complex in the above case, and it is recommended to use the G76 automatic thread cutting cycle if equipped with multiple repetitive cycle. The above calculation is automatically made by the NC in G76.

NOTES ON G92

1. **Designation of the T, S, M functions**

   The T, S, M functions which will act as the cutting conditions of the G92 cycle are basically to be specified before the G92 block. However, T, S, M blocks without move command in the area in which G92 is effective are effective.

2. **G92 cycle when signal-block is "ON"**

   When single-block is "ON", the G92 cycle finishes the 1 to 4 cycle without interruption, and stops after completing the cycle.

3. **Thread cutting hold (option)**

   When the feed hold button is depressed during thread cutting in NCs equipped with this option, thread cutting is rounded off immediately, and it returns to start point A. When parameter #4011 D1 = 1, it stops at position B where the thread cutting rounding off ended.

   Depress the cycle start button at the start point A or the rounding off point B, to start the cutting from the beginning.

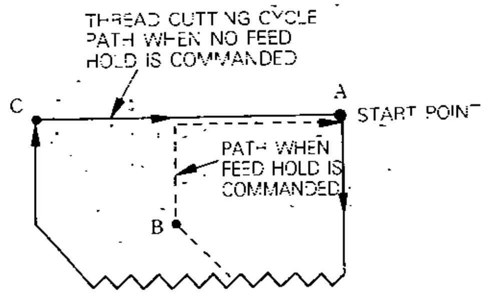


Fig. 2.13.8 Thread Cutting Feed Hold

   When the feed hold button is depressed during thread cutting on NCs without the feed hold option, the thread cutting is continued and does not stop until it ends the clearance at point C.

4. **Infeed of optional angle of thread by G76**

   The thread angle is restricted to 6 types in multiple repetitive cycle: G76, but using this G92 allows thread cutting with optional angle.

5. **When using G92 with chamfering ON**

   When using G92 with chamfering ON, alarm 0454 occurs if the thread rounding off amount is "0".

2.13.1.3 Thread cutting cycle (G94)

(1) Straight thread cutting cycle

| G94 X(U).......Z(W).......F(E)....... ; |

The thread cutting cycle of the following **Fig. 2.13.9** (1 to 4) is executed by the above command.

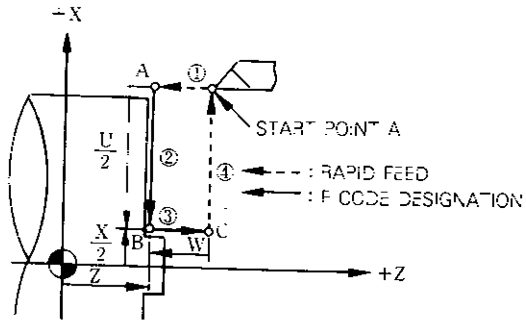


Fig. 2.13.9 Straight Front Thread Cutting Cycle

As G94 is modal, the thread cutting cycle is executed in the following blocks just by commanding infeed in the Z-axis direction by;

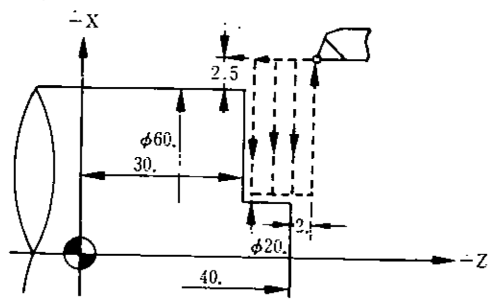
| Z(W)..... ; |
| Z(W)..... ; |

Example of Programming

N60 G00 X65. Z42. ;

N61 G94 X20. Z38. F0.35 ;
N62 Z34. ;
N63 Z30. ;
Cutting 3 cycles by G94

N64 G00............;



(2) Taper thread cutting cycle

G94 X(U).......Z(W).......K........F(E)........;

The above command executes the taper thread cutting cycle (1 to 4) as shown in Fig. 2.13.10.

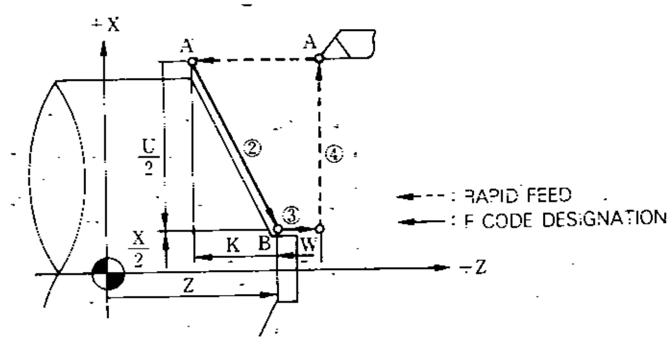


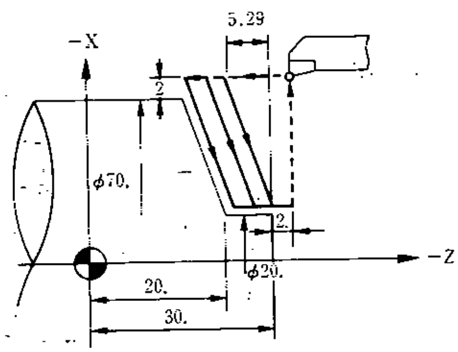
Fig. 2.13.10 Taper Front Thread Cutting Cycle

Specify the sign of K as the direction viewed from point B.

Example of Programming

```
N70  G00  X74.  Z32. ;
N71  G94  X20.  Z30.  K-5.29  F0.3 ;
N72  Z25. ;
N73  Z20. ;
N74  G00············;
```

N71–N73: Cutting 3 cycles by G94



NOTES ON G94

1. **Designation of the T, S, M functions**

   The T, S, M functions which will act as the cutting conditions of the G94 cycle are basically to be specified before the G94 block. However, T, S, M blocks without move command in the area in which G94 is effective are effective.

2. **G94 cycle when single-block is "ON"**

   When single-block is "ON", the G94 cycle finishes the 1 to 4 cycle without interruption, and stops completing the cycle.

2.13.2 Multiple Repetitive Cycle (G70 to G76) *

2.13.2.1 Overview

G90, G92, G94 canned cycles just execute commands equivalent to 4 blocks, which makes the program more complex. The multiple repetitive cycle can be used to significantly shorten the program. For example, rough cutting and finishing cutting can be performed just by commanding the finishing form.

Seven multiple repetitive cycles G70 to G76 are provided, as shown in **Table 2.13.3**.

The G codes of Table 2.13.3 are all non-modal G codes of group * (Table 2.3.3).

Table 2.13.3 G Codes (G70 to G76)

G code	Name	Remarks	
G70	Finishing cycle		
G71	Outer diameter rough cutting cycle	Finishing can be performed by G70.	Nose R compensation can be applied.
G72	End face rough cutting cycle		
G73	Closed loop cutting cycle		
G74	End face cutting off cycle	Nose R compesation can be applied.	
G75	Outer diameter cutting off cycle		
G76	Automatic thread cutting off cycle		

(1) Finishing form program by G71, G72, G73

The finishing form program commanded by G71, G72, G73 is stored in the internal memory of the NC. The maximum capacity is 45 blocks, and a single set of programs can be stored.

$$\boxed{\text{Finishing form program} \leq 45 \text{ blocks}}$$

However, chamfering, rounding (G11, G12), complex chamfering, rounding (G111, G112) must be counted as multiple blocks as shown in **Table 2.13.4**.

Table 2.13.4 Block Count when Using G11, G12, G111, G112

1 G11 or G12 block	Equivalent to 2 blocks
1 G111 block	Equivalent to 4 blocks
1 G112 block	Equivalent to 5 blocks

(2) **Finishing form program storage internal memory**

The finishing form program is stored in a special memory within the NC after binary conversion to shorten the operation time of the rough cutting cycle. This special memory is called the "finishing form program internal memory", which is different from the part program memory.

(3) **Specify G code of group 01 after the G70 to G76 cycle**

Specify the group 01 G code again in the block after the G70 to G76 cycle. This is because execution of the G70 to G76 cycle may change the group 01 G code specified before the cycle to another G code.

(4) **Nose R compesation against G70 to G73**

Nose R compensation can be applied against G70 to G73 cycles.

(5) **Nose R compensation against G74 to G76**

Nose R compensation cannot be applied against G74 to G76 cycles, or a fault will occur.

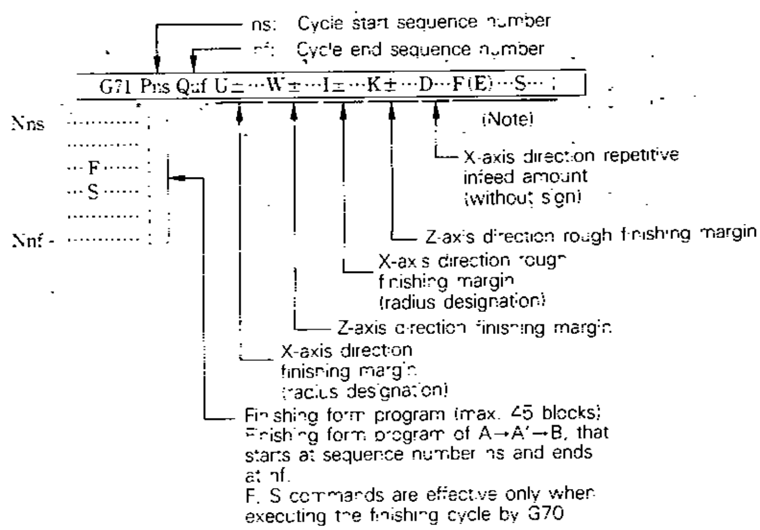
2.13.2.2 Outer diameter rough cutting cycle (G71)

G71 can command rough cutting cycle leaving the finishing margin or a rough finishing cycle on the outer diameter or inner diameter.

The command differs according to when the finishing form is simple increasing/decreasing form, and when it includes dents.

(1) When the finishing form is simple increasing/decreasing form

(a) The cycle shown in **Fig. 2.13.12** is executed by the sequential command of **Fig. 2.13.11**.



NOTE Give the feed command [F(E)] and spindle command (S) used when executing this outer diameter rough cutting cycle.

Fig. 2.13.11 Cycle Command

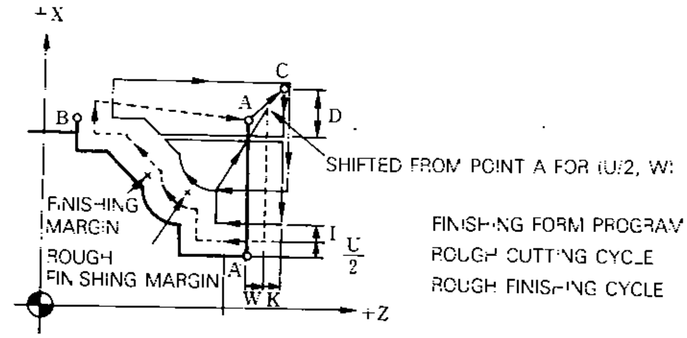


Fig. 2.13.12 Execution of Cycle

Starting from point A, the rough cutting cycle (——) and rough finishing cycle (......) are executed, and it returns to point A.

When I = 0, K = 0, (or no designation), it ends omitting the rough finishing cycle, as shown in **Fig. 2.13.13**.

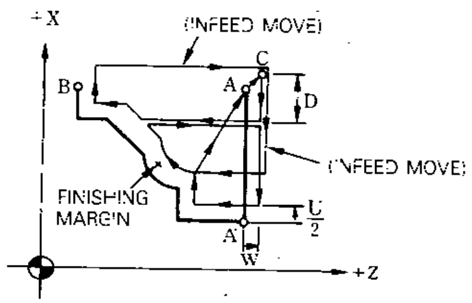


Fig. 2.13.13 Omitting the Rough Finishing Cycle

The "return move" is performed in rapid feed by G00. The speed of the "infeed move" is the speed specified by the AA' program (G00 or G01).

Each infeed amount D in the X-axis direction can be overridden in 10% steps of the setting within the range of 0 to 200%. See note 6 of the G71 notes (page 199)

NOTE 1. The finishing form program can take simple increase or simple decrease forms only.

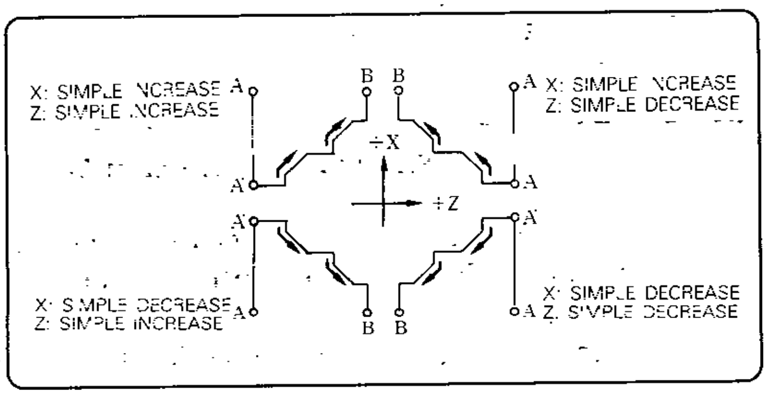


Fig. 2.13.14 Simple Increase, Simple Decrease

2. The start block (Nns....) and end block (Nnf....) of the finishing form program have the following restrictions.

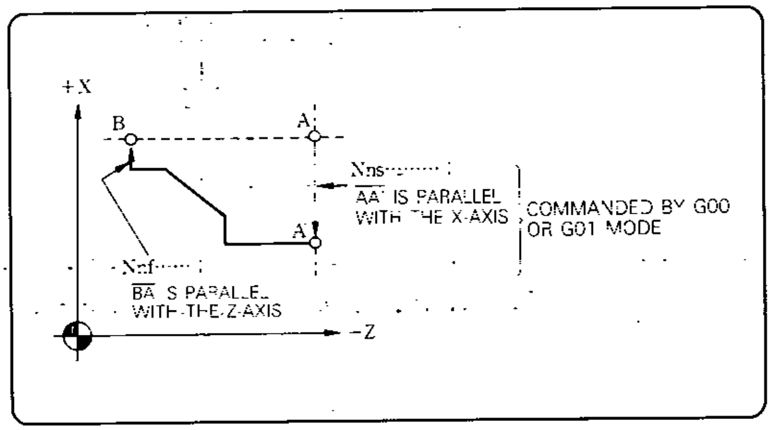
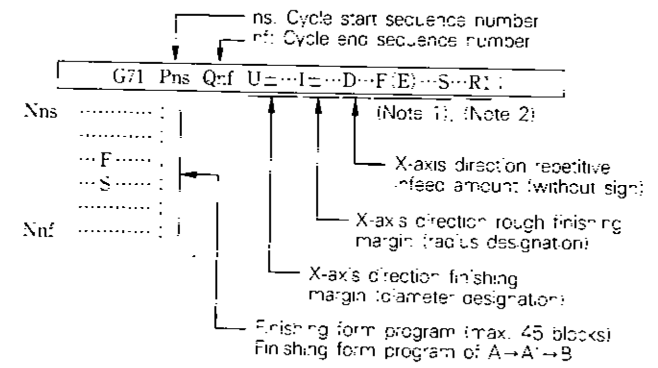


Fig. 2.13.15 Restrictions of Start Block and End Block

(2) When the finishing form has dents

(a) The command of **Fig. 2.13.16** executes a cycle as shown in **Fig. 2.13.17**.



NOTES
1. Give the feed command [F(E)] and spindle command (S) used when executing this outer diameter rough cutting cycle.
2. When "R1" is specified, the cutting path against the finishing form program with dent is operated.

Fig. 2.13.16 Cycle Command

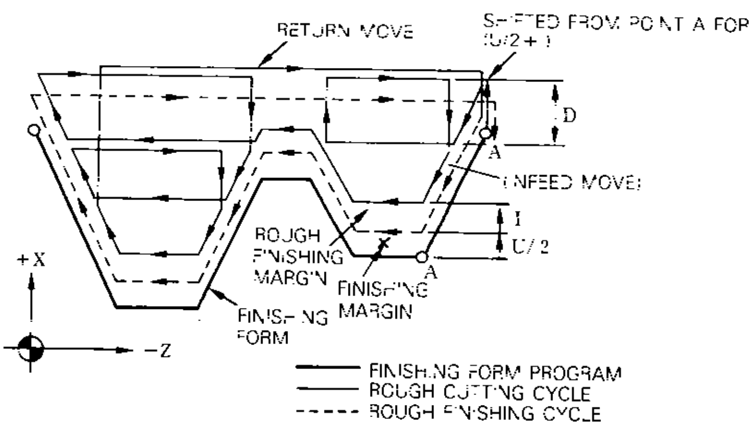


Fig. 2.13.17 Execution of the Cycle

Starting from point A, the rough cutting cycle (——) and rough finishing cycle (---) are executed, and it returns to point A to the end. When I is not specified, it ends omitting the rough finishing cycle.

The "return move" is made in rapid feed by G00. The speed of the "infeed move" is the speed specified by the $\overline{AA'}$ program (G00 or G01).

Each infeed amount D of the X-axis direction can be overridden in 10% steps of the setting within the range of 0 to 200%. See Note 6 of "NOTES ON G71".

NOTES 1. The G71 type B rough cutting cycle starts from the dent nearest the start point.

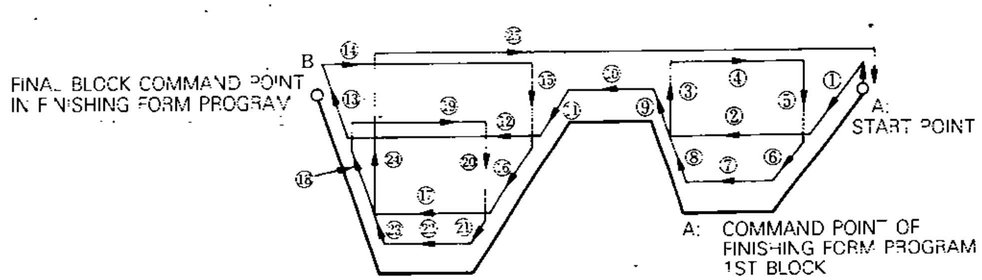


Fig. 2.13.18 Rough Cutting Cycle by G71 Command Type B

As the cutting starts from the dent nearest the start point, the cutting line may hit the peak on the other side of the dent, as shown in **Fig. 2.13.19**. If this happens, the cutting line is interrupted, the cutting cycle is continued to the end of the dent, and it returns to the point where the cutting was interrupted, to start the cutting from the interruption point.

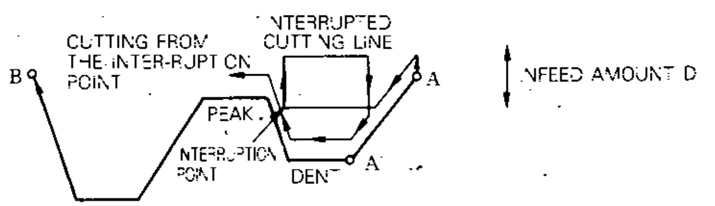


Fig. 2.13.19 Cutting in Case the Cutting Line Hits the Peak at the Other Side of the Dent

NOTES 2. If the dent is minor, there will be a single interruption point, but if there are dents and peaks inside the dent as shown in **Fig. 2.13.20**, there will be more interruption points after the first interruption of the cutting line.

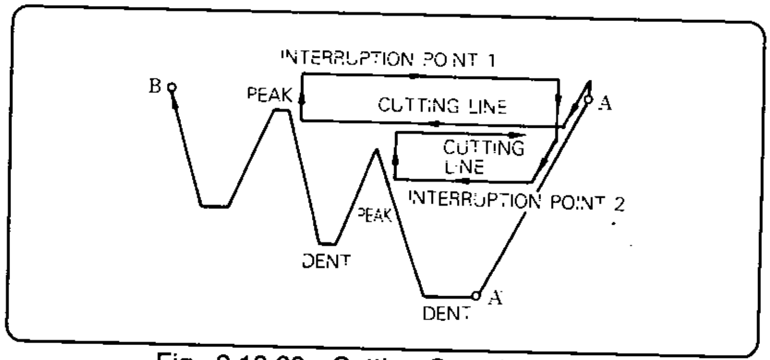


Fig. 2.13.20 Cutting Complex Dents

As showm above, when there are interruption points until the cutting line returns to the first interruption point, cutting can be performed normally if the interruption points are 3 or less, but alarm "0469" occurs if there are 4 or more interruption points.

There is no restriction against the number of dents as long as the interruption points are within the above limits.

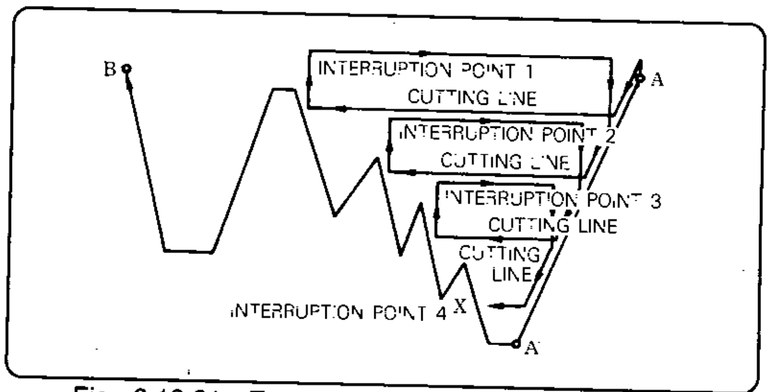


Fig. 2.13.21 Example of Form that cannot be Cut

3. Forms with overhang (projection) cannot be cut. Therefore, the Z-axis command value of the finishing form program must have simple change.

4. The end block of the finishing form program has restrictions as shown in **Fig. 2.13.22**.

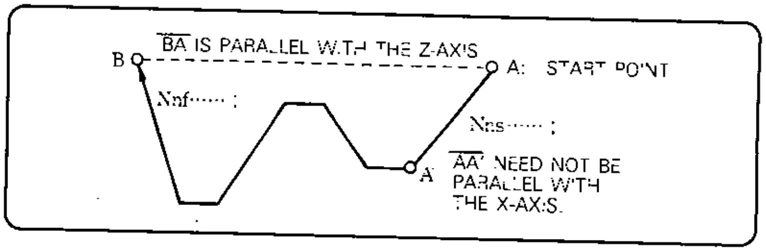


Fig. 2.13.22 Restrictions of the End Block of the Finishing Form Program

Command the G01 or G00 commands in the end block (Nnf...... ;).

NOTES 5. The clearance after each infeed cycle can be set by parameter.

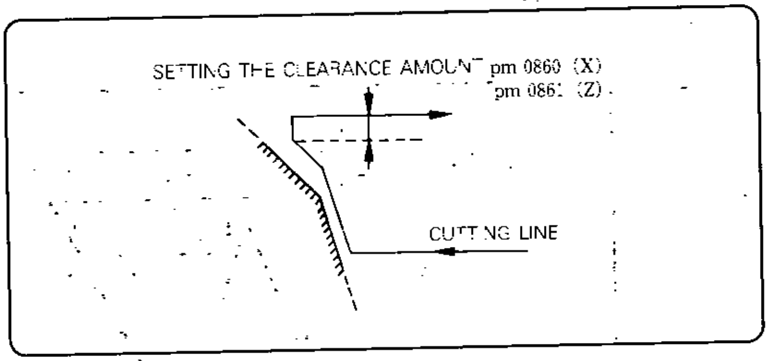


Fig. 2.13.23 Setting the Clearance Amount

6. Each block of the finishing form program must be a simple increase or simple decrease. Arcs that enter multiple quadrants must be divided and programmed in 2 blocks.
7. The finishing margin W, K of Z-axis direction is generally not commanded. If commanded, it may cut into one side of the wall.
8. Unless "R1" is commanded, the rough cutting cycle remains the conventional simple increase/simple decrease cycle.
9. Approach is made at cutting feed. It is not affected by the G code of the finishing form program.

   Therefore, in some finishing form programs, position may be made in rapid feed after approach at cutting feed.

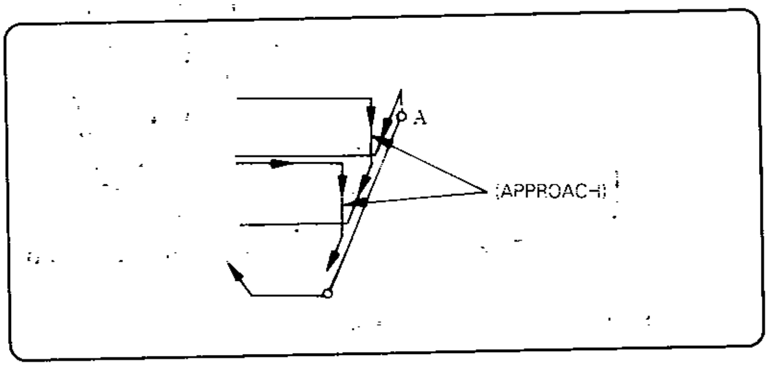


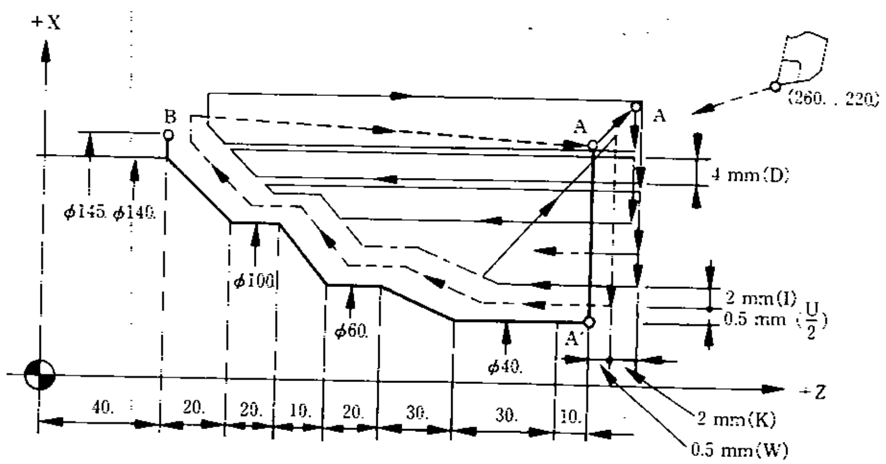
Fig. 2.13.24 Approach

10. When the end point of simple decrease and dent form finishing program is lower than the start point at pm4026 D1=1, the tool may interfere the workpiece.

Example of Programming

When nose R compensation is commanded in finishing form without dents

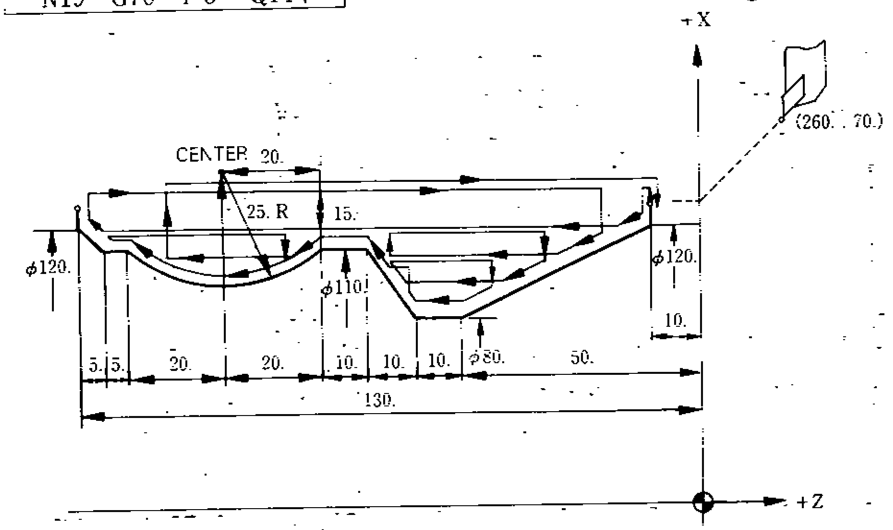
```
N 1   G50   X260.  Z220. ;
N 2   G00   S1000   M03   T0101 ;
N 3   G42 ;
N 4   X145.  Z180. ;                        Outer diameter rough
                                            cutting cycle
N 5   G71  P6   Q13   U1.  W0.5  I2.  K2.  D4.  F0.3  S800 ;
N 6   G00   X40.  S800 ;<--Rapid feed infeed
N 7   G01   W-40.  F0.15 ;
N 8   X60.  W-30.  S600 ;
N 9   G12   W-20.  I5.  ;<--Equivalent to
                            2 blocks            Finishing form=9 blocks
N10   G01   X100.  W-10.  S300 ;
N11   W-20. ;
N12   X140.  W-20.  S200 ;
N13   X145. ;
N14   G40 ;
N15   G00   X260.  Z220.  T0100 ;
```



Example of Programming

When nose R compensation is not commanded in finishing form with dents

```
N01  G50  X260.  Z70.  ;
N02  G00  S500  M03  T0101 ;
N03  X124.  Z-10.  ;                         Outer diameter rough
                                             cutting cycle
N04  G71  P5  Q14  U2.  D6.  F0.2  S250  R1 ;
N05  G01  X120.  ;
N06  X80.  Z-50.  F0.1  S500 ;
N07  W-10.  ;
N08  X110.  W-10.  ;
N09  W-10.  ;                                Finshing form=
N10  G02  X90.  W-20.  I15.  K-20.  ;        9 blocks
N11  X110.  W-20.  I25.  ;
N12  G01  W-5.  ;
N13  X120.  W-5.  ;
N14  X124.  ;
N15  G00  X260.  Z70.  T0100 ;
N16  T0202 ;
N17  G50  X255.  Z70.  ;
N18  X124.  Z-10.  ;
N19  G70  P5  Q14 ;                          Execution of the G71 finishing
```



NOTES ON G71

1. **Address U, W, I, K, D commands**

   Address U, W, I, and K are commanded with sign. Incorrect sign can cause excess infeed.

   Repetitive infeed amount D is specified without sign.

2. **Command the finishing form program immediately after the G71 block. The program is disregarded if written between programs.**

3. **F, S, T code commands**

   When no F, S, T codes are found in the G71 block, the F, S, T codes specified in the previous blocks are applied to the outer diameter rough cutting cycle.

   The F, S < T, codes commanded in the finishing form program become effective only in the finishing cycle (G70), and are disregarded during the outer diameter rough cutting cycle.

4. **G codes that can be specified in the finishing form program**

   Table 2.13.5 show the G codes that can be specified in the finishing form program, except for the Nns, Nnf blocks.

Table 2.13.5 G Codes that can be Used

G Codes that can be Used	Remarks
G00, G01, G02, G03, G32, G23, G41, G42	
G11, G12	Calculated as 2-block equivalent
G111	Calculated as 4-block equivalent
G112	Calculated as 5-block equivalent

5. **Applying nose R compensation**

   Nose R compensation is effective in the G71 cycle when it has already entered the nose R compensation mode before giving the G71 command. However, note that;

Nose R compensation is not executed in the rough cutting cycle
Nose R compensation is executed in the rough finishing cycle

   Therefore, nose R compensation is disabled in programs where the rough cutting cycle is omitted (I = 0, K = 0).

   Nose R compensation can be executed in the G70 to G73 cycles, also in the finishing form program.

   Thus, the G41 and G42 are added to the G codes that can be specified in the finishing form program (except for the Nnf, Nnf blocks). Nose R compensation is enabled in the rough finishing cycle and finishing cycle from the block where G41 or G42 is commanded.

   Always command G41/G42 in the finishing form program start block, with G00 or G01 move command. The command cannot be given in a single block.

6. **G71, G72 infeed amount override**

   The repetitive infeed amount D can be overridden in 10% steps within the range of 0 to 200% as follows.

   Set the setting parameter pm0023 D0 to D4 (in 5-bit code)

7. **Rough finishing margin against the G70 to G73 cycles**

   When omitted by I, K, the finishing margin U, W can be executed as rough finishing margin. (Enabled by parameter pm4026 D0 = 1)

8. **G72 cycle**

   All the above notes on the G71 command are also applied to the G72 cycle. The cutting by G72 is performed as a move parallel with the X-axis, but all other details are the same as G71.

2.13.2.3 End face rough cutting cycle (G72)

G72 commands the rough cutting cycle and rough finishing cycle leaving the finishing margin. While the G71 cutting is performed parallel with the Z-axis, the G72 cutting is performed parallel with the X-axis.

(1) When the finishing form is simple increase/decrease

(a) The commands of **Fig. 2.13.25** execute a cycle as shown in **Fig. 2.13.26**.

```
G72 Pns Qnf U±……W±…I±…K±…D…F(E)…S…;

Nns…………………;
…………………;
F…………;
S…………;
…………………;
Nnf…………………;
```

- U, W, I, K: The meaning of the address is the same as G71
- D: Z-axis direction repetitive infeed amount (without sign)
- Nns to Nnf: Finishing form program (max. 45 blocks) Form program of A→A'→B

Fig. 2.13.25 Cycle Command

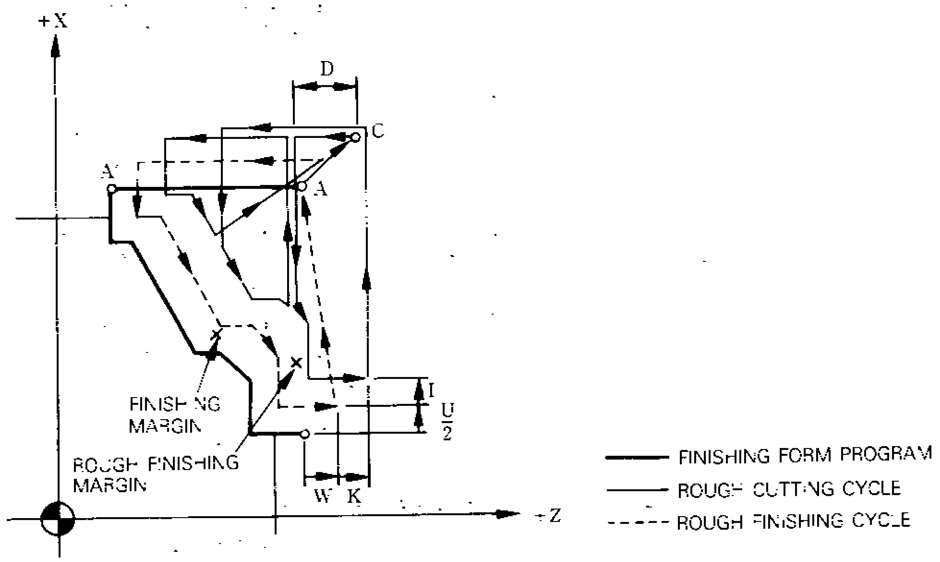


Fig. 2.13.26 Execution of the Cycle

Starting from point A, the rough cutting cycle (——) and rough finishing cycle (---) are executed, and it returns to point A to end.

When I = 0, K = 0 (or no designation), it ends omitting the rough finishing cycle.

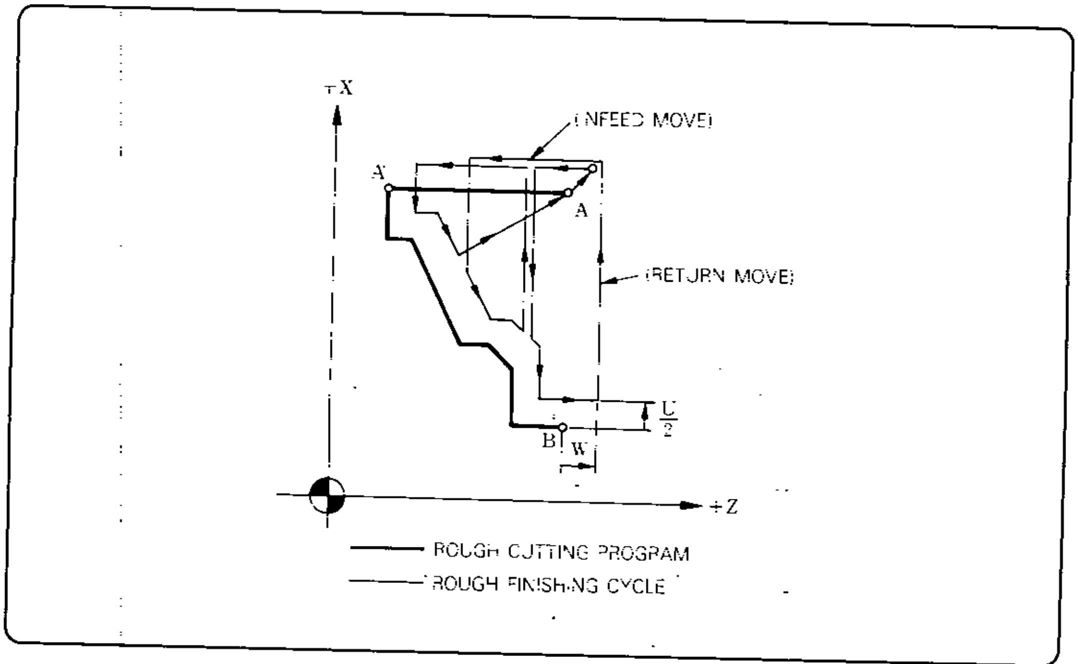


Fig. 2.13.27 Execution of the Cycle when I = 0, K = 0

The "return move" is made in rapid feed by G00. The speed of the "infeed move" is the speed specified by the $\overline{AA'}$ program (G00 or G01).

The repetitive infeed amount D in the Z-axis direction can be overridden in 10% steps of the setting within the range of 0 to 200%.

NOTES 1. The above notes on the G71 command are also applied to the G72 cycle. The cutting by G72 is performed as a move parallel with the Z-axis, but all other details are the same as G71.

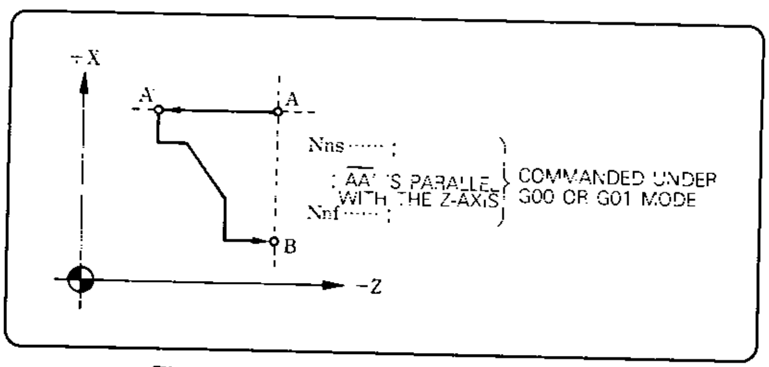
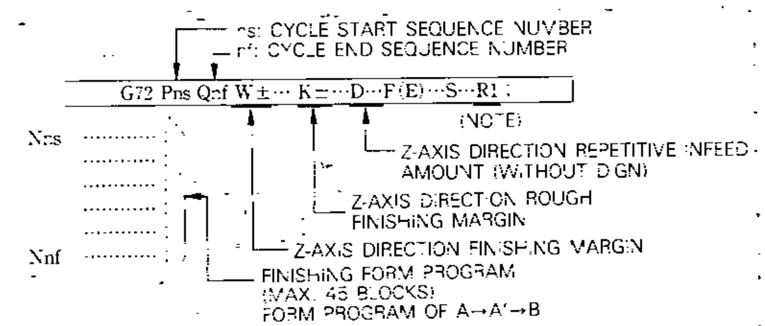


Fig. 2.13.28 Infeed Direction in G72

(2) When the finishing form has dents



NOTE When "R1" is specified, the cutting path is operated against the finishing form program with dents.

Fig. 2.13.29 Cycle Command

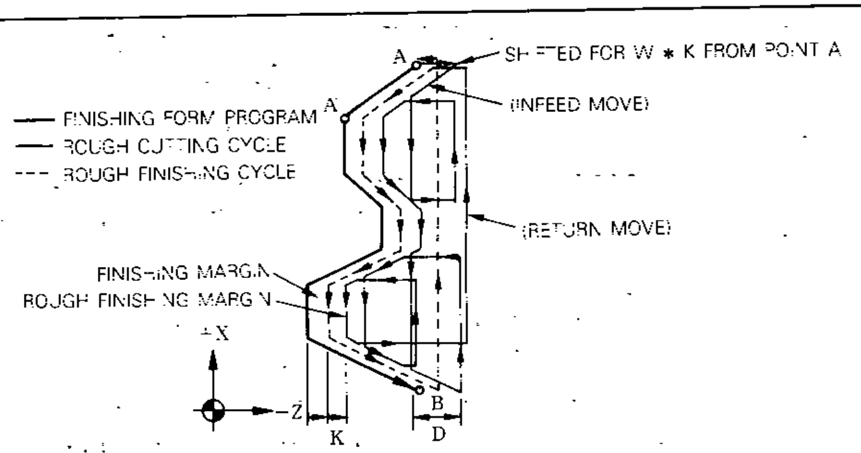


Fig. 2.13.30 Execution of the Cycle

Starting from point A, the rough cutting cycle (——) and rough finishing cycle (---) are executed; and it returns to point A to end. When K is not specified, the rough finishing cycle is omitted.

The "return move" is made in rapid feed by G00. The speed of the "infeed move" is the speed specified by the $\overline{AA'}$ program (G00 or G01).

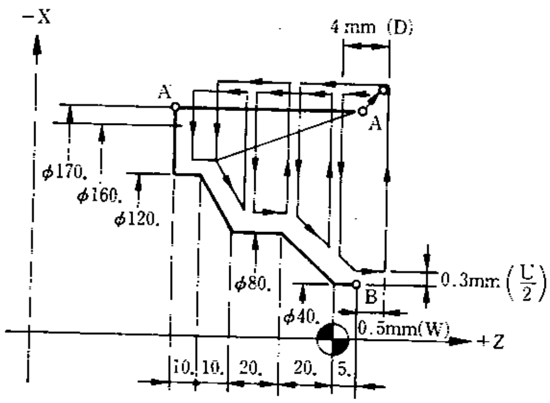
The repetitive infeed amount D to the X-axis direction can be overridden in 10% steps of the setting within the range of 0 to 200%.

NOTE The above notes on the G71 command are also applied to the G72 cycle. The cutting by G72 is performed as a move parallel with the Z-axis, but all other details are the same as G71. However, the "G72 clearance" can be set by the setting parameters #0863 (Z) and #0862 (X).

NOTE ON G72
The notes on the G71 and G72 commands are completely the same.

Example of Programming

When nose R compensation is not commanded by I = K = 0



N1 G50 X260. Z60. ;
N2 G00 S1000 M03 T0202 ;
N3 X170. Z5. ; — End face rough cutting cycle

N4 G72 P5 Q11 U0.6 W0.5 I0 K0 D4.0 F0.3 S200 ;
N5 G01 Z-60. F0.15 ; ···Infeed is performed by cutting feed
N6 X120. S250 ;
N7 Z-50.
N8 X80. Z-40. S400 ;
N9 Z-20. ;
N10 X40. Z0 S800 ;
N11 Z5. ;

(N5–N11: Finishing form program)

N12 G00 X260. Z60. ;
N13 T0303 ;
N14 X170. Z5. ;
N15 G70 P5 Q11 ; ← The finshing cycle is exexuted.

〈Remarks on G71, G72〉

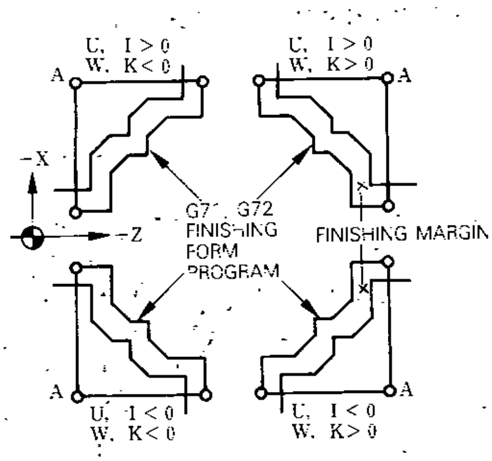


Fig. 2.13.31 G71, G72 Finishing Form Program and Signs of Addresses U, W, I, K

NOTE If the sign of U, W, I, or K is incorrect, excess infeed as shown in **Fig. 2.13.32** occurs.

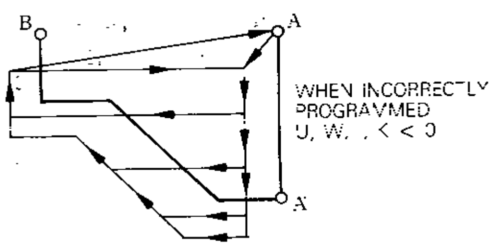
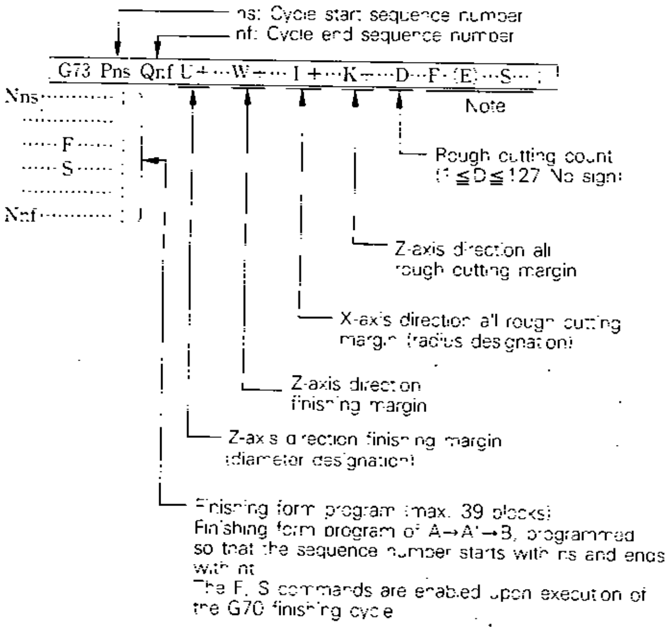


Fig. 2.13.32 Infeed when the Sign of U, W, I, or K is Incorrect

2.13.2.4 Closed loop cutting cycle (G73)

G7 is a command effective against a workpiece with a finishing form similar to casting or forging forms.

(1) A cycle as shown in **Fig. 2.13.34** is executed by the commands of **Fig. 2.13.33**.



NOTE Give feed command (F(E)) and spindle command (S) used in this closed loop cutting cycle.

Fig. 2.13.33 Cycle Command

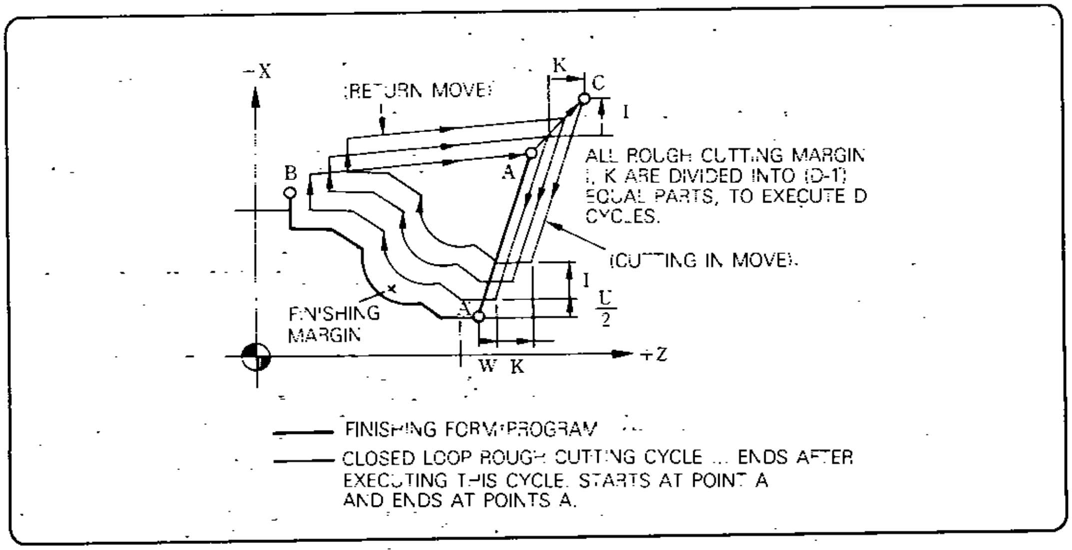


Fig. 2.13.34 Cycle Execution

The "return move" is performed by G00.

The "cutting in move" speed is the speed specified in the program of $\overline{AA'}$.

NOTE

1. Command addresses U, W, I, K with sign.
2. The rough cutting count D is to be specified without sign, with the following restriction.

   $\boxed{1 \leq D \leq 127}$

   Alarm occurs when it exceeds this value. (Alarm 0467). When D = 1 is commanded rough cutting of I, K is performed in a single cycle, leaving the finishing margin.
3. The finishing form must be commanded immediately after the G73 block.
4. Command the finishing form program start block (Nns...) and the end block (Nnf....) with G00 or G01.
   - It need not be parallel with the axes.
5. The finishing form program need not be a simple increase or simple decrease.

Example of Programming

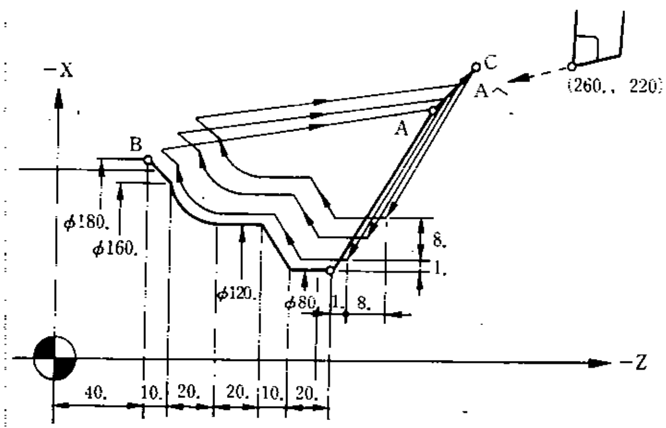
```
N10  G50  X260.  Z220. ;
N11  G00  S300  M03  T0303 ;
N12  X220.  Z160. ;
```

Closed loop cutting cycle

```
N13  G73  P14  Q19  U2.  W1.  I8.  K8.  D3  F0.3  S200 ;
N14  G00  X80.  W-40.  S400 ;
N15  G01  W-20.  F0.15 ;
N16  X120.  W-10.  S300 ;
N17  W-20. ;
N18  G02  X160.  W-20.  R20.  S200 ;
N19  G01  X180.  W-10. ;
```

N14–N19: Finshing form program

```
N20  G00  X260.  Z220. ;
```



NOTES ON G73

1. **When there is no F(E), S code designation within the G73 block**

   When there is no F(E), S code designation within the G73 block, the F(E), S codes specified in the preceding blocks are used in the closed loop cutting cycle.

   The F(E), S, T codes specified in the finishing form are enabled in the finishing cycle (G70), and disregarded in the closed loop cutting cycle.

2. **G codes that can be specified in the finishing form program**

   Table 2.13.7 show the G codes that can be specified in the finishing form program, except for the Nns, Nnf blocks.

   Table 2.13.7 G Codes that can be Used

G Codes that can be Used	Remarks
G01, G00, G02, G03, G22, G23, G41, G42	
G11, G12	Calculated as 2-block equivalent
G111	Calculated as 4-block equivalent
G112	Calculated as 5-block equivalent

3. **When both rough cutting margins I and K are "0" or not specified, input fault occurs (alarm 0467).**

4. **Rough cutting margin ΔI, ΔK**

   Each rough cutting margin ΔI, ΔK is calculated as follows:

   $$\Delta I = \frac{I}{D-1}, \quad \Delta K = \frac{K}{D-1} \quad \text{with } D \leqq 2$$

   The NC omits digits below 0.001.
   Do not specify ΔI and ΔK with values less than 0.001 mm in the program.

   (Remark) Example of ΔI, ΔK fractions

   (Example 1) When I = 0.005 mm, K = 0.005 mm, D = 7,

   $$\left.\begin{array}{l}\Delta I = \dfrac{0.005}{6} = 0 \\ \Delta K = \dfrac{0.005}{6} = 0\end{array}\right\}$$ Therefore, it causes the Note 3 fault.

   (Example 2) When I = 0.01 mm, K = 0.01 mm, D = 7,

   $$\left.\begin{array}{l}\Delta I = \dfrac{0.1}{6} = 0.001 \text{ mm} \\ \Delta K = \dfrac{0.01}{6} = 0.001 \text{ mm}\end{array}\right\}$$

   Therefore, the margin in each cycle is executed as follows.
   1st to 5th cycle ...... ΔI = ΔK = 0.001 mm
   7th cycle .... ΔI = ΔK = 0.004 mm

5. ***Applying nose R compensation**

   Nose R compensation is executed on all the G73 cycles when it has already entered the nose R compensation mode before giving the G73 command.

2.13.2.5 Finishing Cycle (G70)

After rough cutting is performed by the above G71, G72, G73, finishing cutting is performed by the command following G70.

(1) G70 Pns Qnf ;

nf: Finishing cycle ending sequence number

ns: Finishing cycle starting sequence number

The above command executes only the "finishing form program" of the G71, G72, or G73 cycle.

The F (E), S, T codes specified in the finishing form program are effective during execution of the finishing cycle G70.

The rough cutting F (E), S, T codes specified in the G71, G72, G73 block are disabled in the finishing cycle.

NOTE: The finishing form program in the finishing form program storage internal memory is converted into binary notation before being stored in the NC, so as to shorten the rough cutting cycle operation time. This memory is called the "finishing form program memory".

(2) The G70 command need not be given immediately after the G71, G72, or G73 cycle commands. For example, commands to change the rough cutting tool to the rough finishing tool can also be inserted.

However, commands or operations as shown in **Table 2.13.8** may not be inserted.

Table 2.13.8 Prohibited Command or Operation

Prohibited Command or Operation	Result
"M02", "M30" for internal reset	The "finishing form program" in the memory will be erased
Reset	

(3) Storage of the finishing form program and search function

Different kinds of processing is performed for the tape operation mode and memory operation mode of the finishing form program.

(a) Tape operation mode (TAPE)

```
┌ G71  Pns  Qnf.... ; ← Outer diameter rough cutting cycle
│                       command
│ Nns...... ;  ─┐
│   ⋮           ├── Finishing form program (A)
└ Nnf...... ;  ─┘
  G70  Pns  Qnf.... ; ← Execution of finishing by program (A)
┌ G72  Pns'  Qnf'.... ; ← End face rough cutting cycle command
│ Nns'...... ;  ─┐
│   ⋮            ├── Finishing form program (B)
└ Nnf'...... ;  ─┘
```

When the above command is executed, the finishing form program (A) is erased, leaving the form program (B) in the internal memory.

Thus, later finishing by G70 is commanded against the finishing form program (B).

Alarm ("0462") occurs if the sequence number specified by G70 and the sequence number in the finishing form program memory do not match.

(b) Memory operation mode (MEM)

If the sequence number specified by G70 and the sequence number in the finishing form program memory match, the finishing cycle is executed normally. However, if they do not, the specified finishing form program is searched for from the part program, and stored in the internal memory before execution. This function is called the "finishing form program search function".

In this function, two or more rough cutting cycles (or closed loop cutting cycle) can be programmed to be executed in the memory operation as shown below, before performing each finishing cycle.

```
Rough cutting cycle (A)
        ↓
Rough cutting cycle (B)
        ↓
Finishing cycle (A)
        ↓
Finishing cycle (B)
```

⟨Supplement⟩

① Programs using the "finishing form program search function" take longer cycle time than when not using the function. The following are some examples.

☆ **Finishing form program search time**

Example

When the tape length is 5 m (200 characters) and the finishing form program locates in the center, the search time is approx. 100 ms.

☆ **Execution time to store the seached finishing form program in the internal memory**

Example

When the finishing form program has 30 blocks, the execution time is approx. 1 s.

Note that storage of the memory search finishing form program to the internal memory is performed in the block before the G70 command block.
Therefore, extension of the cycle time is the above time minus execution time of the G70 command previous block.

② This search function is executed within the part program with the program number where the G70 command is given.

NOTES ON G70

1. Input error occurs when the finishing cycle start and end sequence numbers ns, or nf are as follows.
   - When the sequence numbers ns, or nf specified by G70 and the finishing form program memory sequence number do not match (tape operation only)
   - When the sequence number nf specified by G70 is commanded before ns or when ns = nf in the finishing form program.
2. When it entered nose R compensation mode before the G70 command is given, nose R compensation is executed against the G70 finishing cycle.

2.13.2.6 End face cutting off cycle (G74)

G74 can command the end face cutting cycle while performing pecking parallel to the Z-axis.

(1) The **Fig. 2.13.35** command execution a cycle results in the function as shown in **Fig. 2.13.36**.

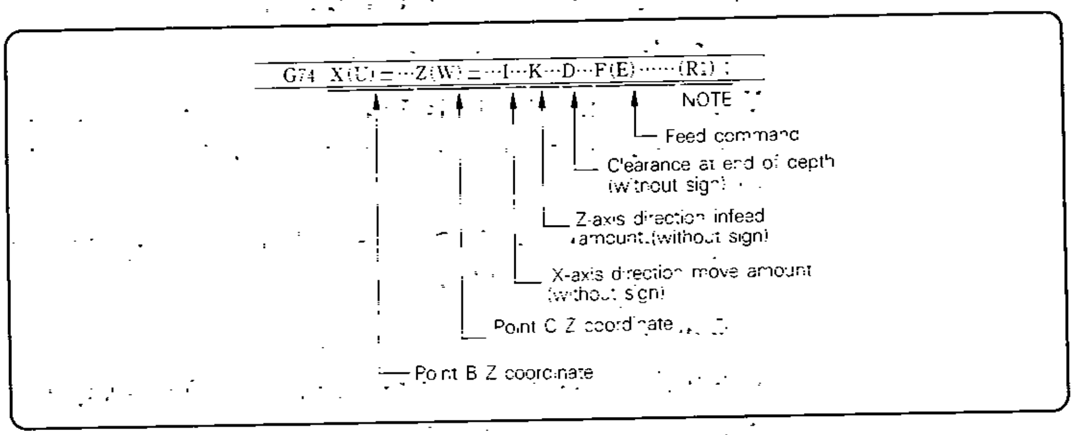


Fig. 2.13.35 Cycle Command

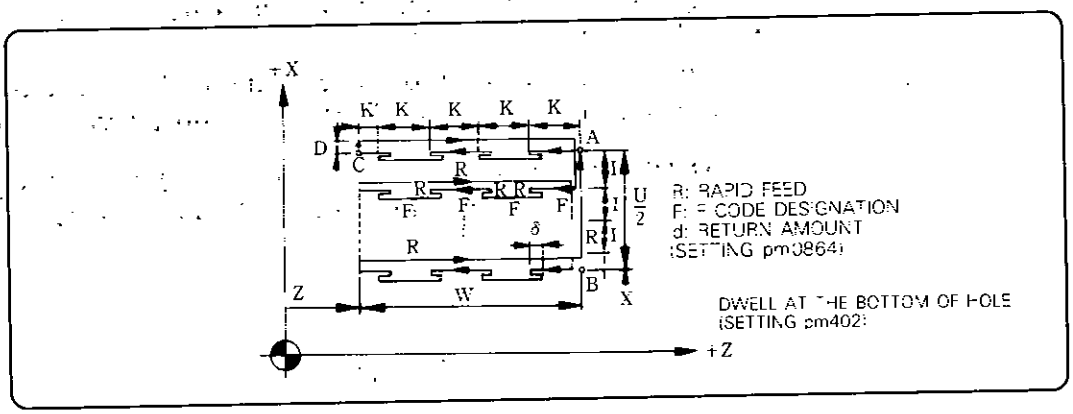


Fig. 2.13.36 Cycle Execution

NOTE The above figure is when there is no "R1" command. If "R1" is commanded, the return amount (d) for each infeed is disregarded, to return the tool to the infeed start point or Z-axis direction point A level each time.

The cycle starts at point A and ends at point A.

(2) Grooving canned cycle considering the infeed steps and tool width can be executed by commanding A, or B as shown below with G74.

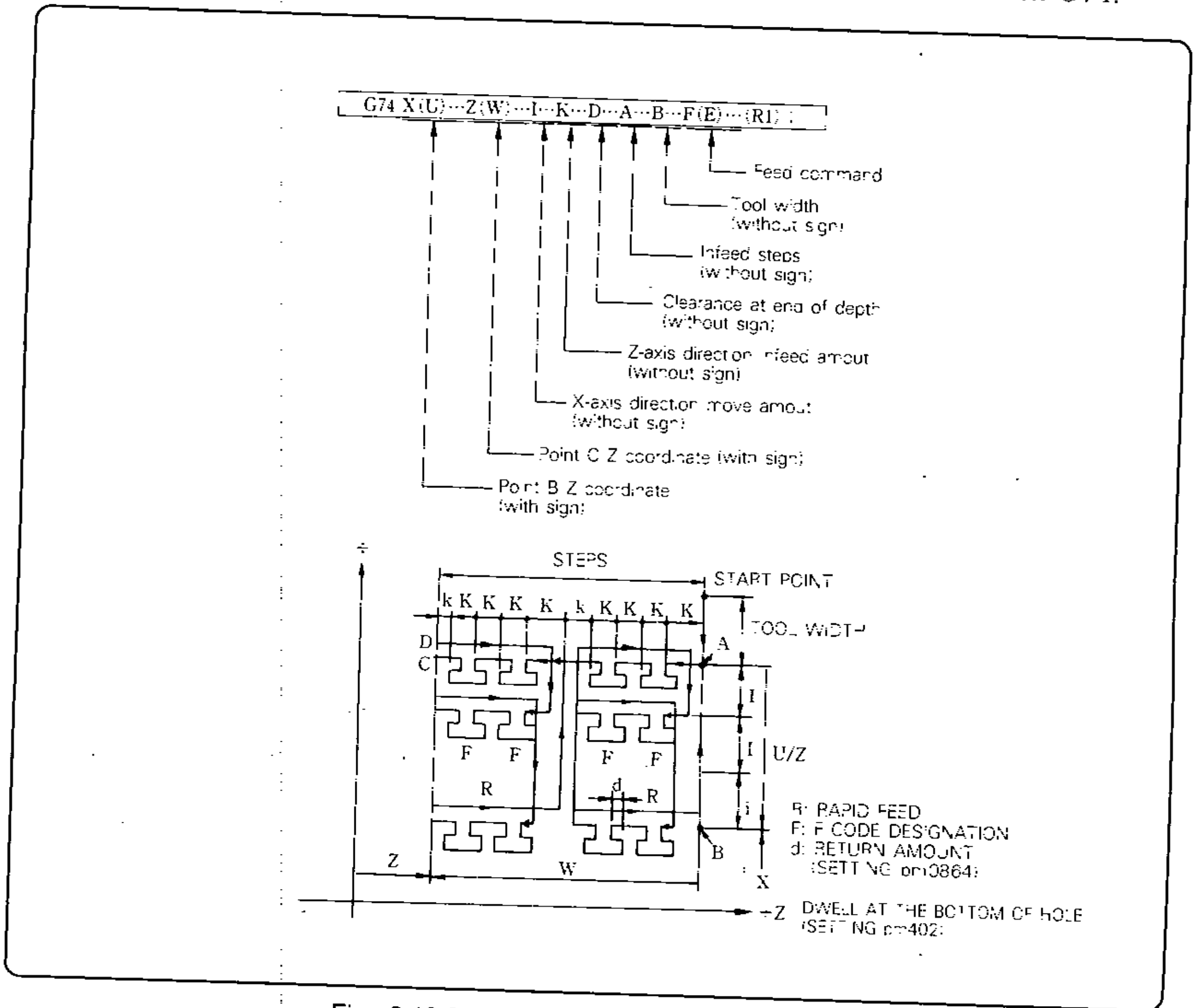


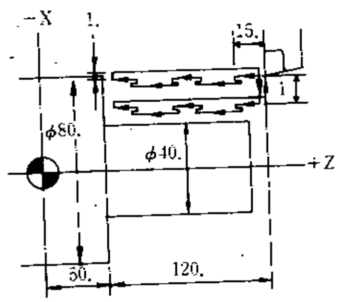
Fig. 2.13.37 G74 Grooving Canned Cycle

NOTES

1. When there is neither command A nor B, the execution is the same as the ordinary G74/G75.
2. When B alone is commanded, the tool width shift is executed at the beginning and end of the G74/G75.
   (a) The first movement is the shift for the tool width from the positioning location at the block immediately before G74/G75 to the X-axis command direction for G74 and to the Z-axis command direction for G75.
   (b) The last move is positioning at a location shifted from the positioning location of the block immediately before G74/G75 for the tool width, and returned to the position of the block immediately before G74/G75.
3. When A alone is commanded, only grooving is executed without the tool width shift.
4. When A command is given, the return amount is the setting of pm0867 in G74 and of pm0868 in G75. No pecking is performed if the data are "0".
5. Alarm 0472 occurs when the groove width < B (tool width).

Example of Programming

G74 X40. Z50. I4. K15. D1. F0.25;



NOTES ON G74

1. Specify addresses I, K or D without sign.
2. When I > |U/2| , pecking starts from and ends at point A.
3. When K > |W| , no pecking is performed, and it cuts directly to the bottom.
4. When D = 0 or no designation is given, no allowance is made for clearance at the bottom.
5. The Z-axis direction final infeed amount K' and X-axis direction final infeed amount I' are automatically operated.
6. When X(U), I, or D are not specified, only 1 cycle of the Z-axis is executed. This is used for deep hole drilling.
7. When the setting of pm0864 is "0", no pecking is performed, and it cuts directly to the bottom.
8. Nose R compensation is disabled in the G74 and G75 cycles.

2.13.2.7 Outer diameter cutting off cycle (G75)

G75 can specify the outer diameter cutting off cycle while performing pecking parallel with the X-axis.

(1) The command of **Fig. 2.13.38** executes the cycle of **Fig. 2.13.39**.

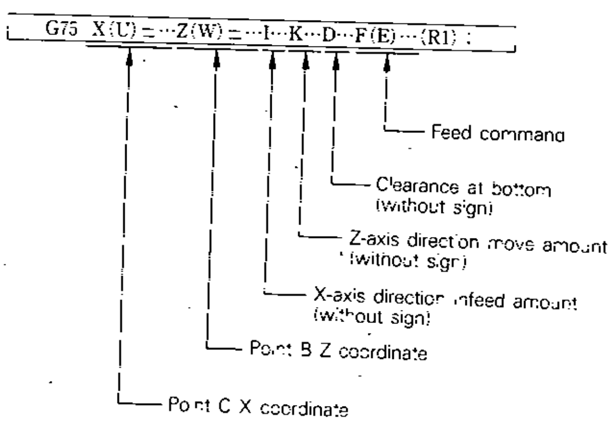


Fig. 2.13.38 Cycle Command

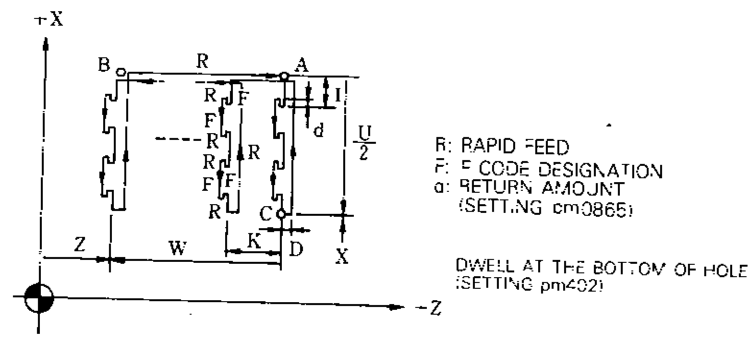


Fig. 2.13.39 Cycle Execution

NOTE The above figure is when there is no "R" command. If "R1" is commanded, the return amount (d) for each infeed is disregarded, to return the tool to the infeed start point or X-axis direction point A level each time.

The cycle starts at point A and ends at point A.

(2) Grooving canned cycle considering the infeed steps and tool width can be executed by commanding A, or B as shown below with G74.

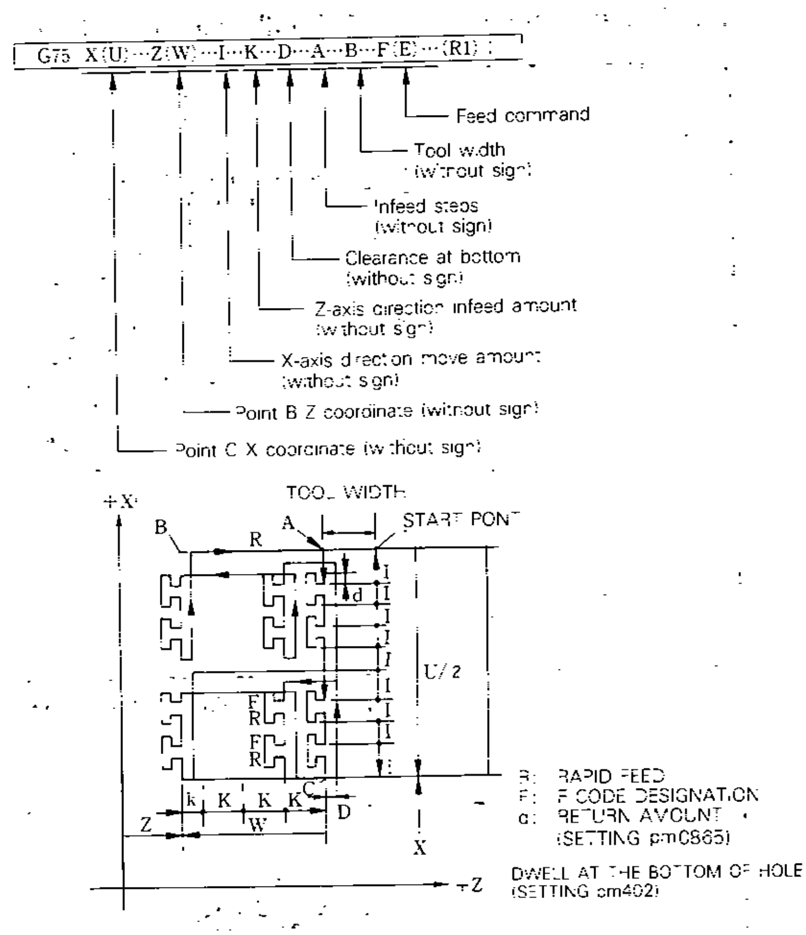
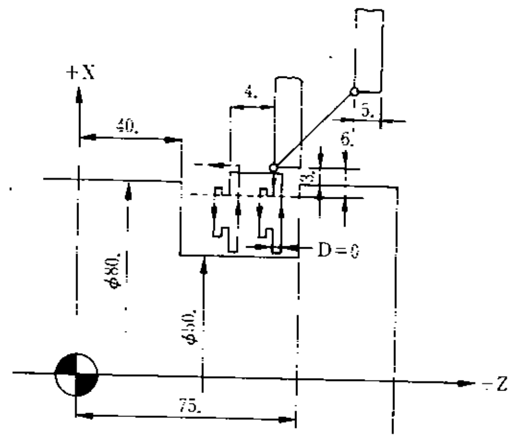


Fig. 2.13.40 G75 Grooving Canned Cycle

NOTES
1. When there is neither command A nor B, the execution is the same as the ordinary G74/G75.
2. When only B is commanded, the tool width shift is executed at the beginning and end of the G74/G75.
   (a) The first movement is the shift for the tool width from the positioning position at the block immediately before G74/G75 to the X-axis command direction for G74 and to the Z-axis command direction for G75.
   (b) The last move is positioning at a location shifted from the positioning location of the block immediately before G74/G75 for the tool width, and returns to the location of the block immediately before G74/G75.
3. When only A is commanded, only grooving is executed without the tool width shift.
4. When A command is given, the return amount is the setting of pm0867 in G74 and of pm0868 in G75. No pecking is performed if the data are "0".
5. Alarm 0472 occurs when the groove width < B (tool width).

Example of Programming

```
N1  G00  X86.  Z70.
N2  G75  X50.  Z40.  I6.  K4.  (D0)  F0.2;
```



NOTES ON G75

Cutting by G74 is performed by parallel move with the Z-axis, while cutting by G75 is performed by parallel move with the X-axis. Therefore, all the notes of G75 are the same as G74, only the direction is different. See the notes on G74.

2.13.2.8 Automatic thread cutting cycle (G76)

G76 commands the automatic cycle for straight thread cutting where infeed is performed along the angle of the thread, and taper thread cutting.

(1) The command of **Fig. 2.13.41** executes the cycle of **Fig. 2.13.42.**

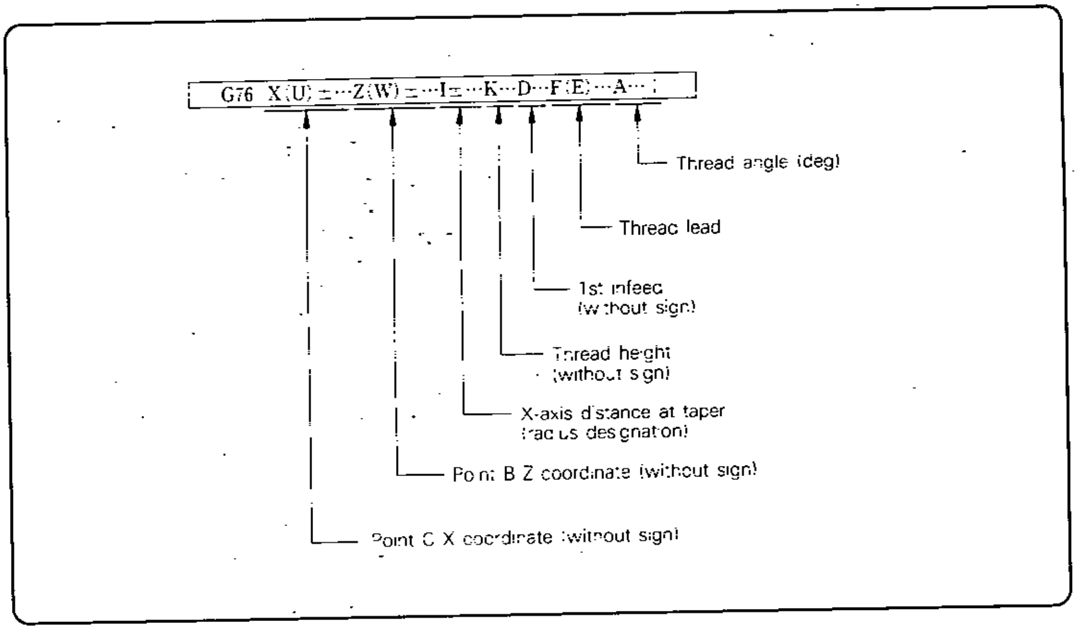


Fig. 2.13.41 Cycle Command

The sign of address I is specified as the direction of viewing point B' from point C.

The cycle start from point A and ends at point A.

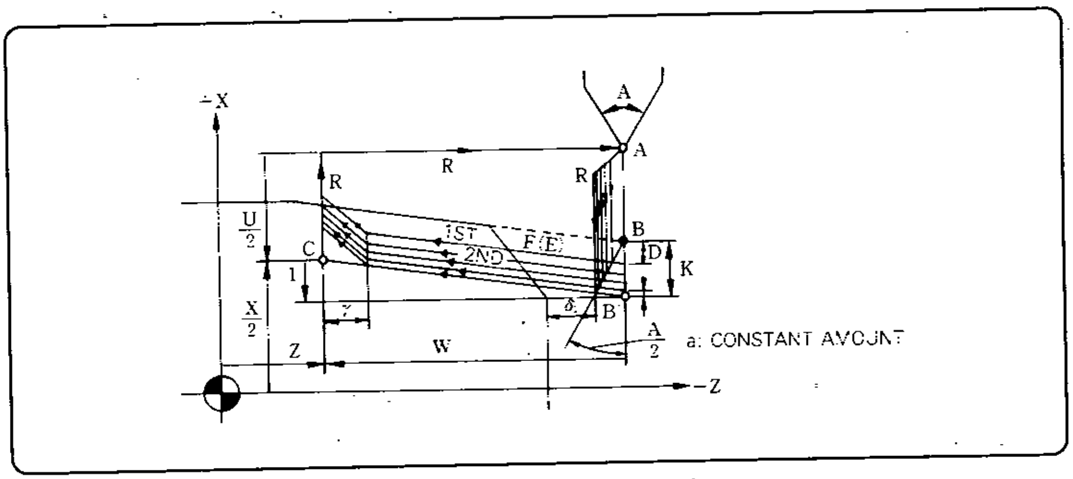


Fig. 2.13.42 Cycle Execution

Infeed near point B is performed as shown in **Fig. 2.13.43**. (Taper thread cutting)

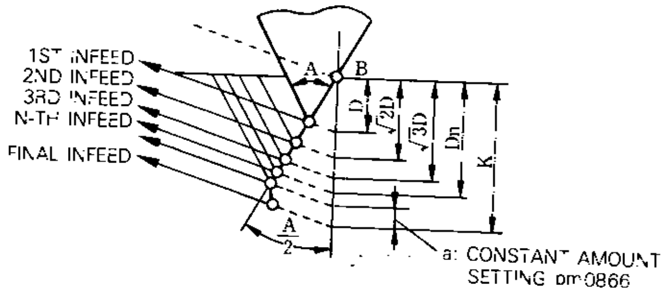


Fig. 2.13.43 Infeed near Point B

The n-th infeed depth Dn is $\boxed{Dn = \sqrt{n}\,D}$.

The thread angle can be selected from the following 6 types.

$\boxed{A = 0°, 29°, 30°, 55°, 60°, 80°}$

The final infeed is performed in the X-axis direction for the constant amount "a". This amount "a" is the value set in SETTING pm 0866.

(2) Straight thread cutting

When address I is "0" or not specified, straight thread cutting as shown in **Fig. 2.13.44** is executed.

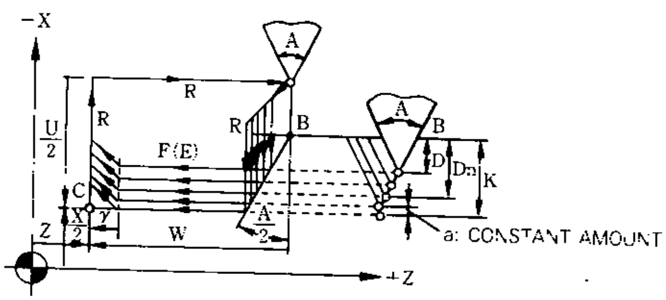
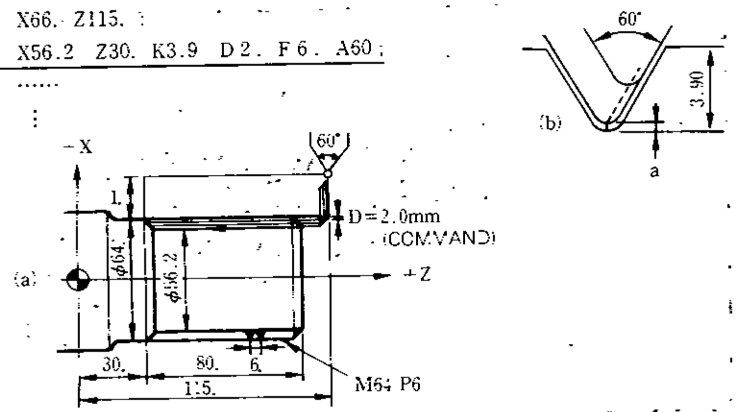


Fig. 2.13.44 Straight Thread Cutting

Example of Programming

```
⋮
G00  X66.  Z115. ;
G76  X56.2  Z30.  K3.9  D2.  F6.  A60 ;
G00  ......
⋮
```



When constant amount a = 0.2 mm, the infeed depth is;

1st infeed ——	1.700 mm	4th infeed ——	3.700 mm
2nd infeed ——	2.528 mm	5th infeed ——	3.900 mm
3rd infeed ——	3.164 mm		

The 1st infeed amount command is D = 2.0 mm, but 1.7 mm infeed is actually performed as result of differential operation of $\sqrt{n}$end D.

(3) Rounding the thread

The thread is rounded when "rounding the thread (CDZ)" is on when the G76 command is given.

The thread rounding amount can be set in parameter pm0100 in 0.1L unit within the range of 0 to 25.5L.

L is the commanded thread lead.

(4) L can be added to the G76 command to execute the last specified n-th cycle.

G76 X(U)±...Z(W)±...I±...K...D...F(E)...A...L... ;

L0 = Execute the command of the final infeed

L1 = Execute from one cycle before the next to the last cycle.

to

Ln = Execute from n cycle before the next to the last cycle.

When n is larger than the normal cutting cycle count (N), (n ≧ N), normal cutting cycle is executed.

(5) The P command can be added to the G76 command for constant amount zig-zag cutting.

G76 X(U) ±...Z(W)±...I±...K...D...F(E)...A...P.. .;

The following infeed can be according to the commanded P.

No P: Constant cutting amount, half-side cutting
P1 : Constant cutting amount, half-side cutting
P2 : Constant cutting amount, zig-zag cutting
P3 or more: Constant cutting amount, half-side cutting

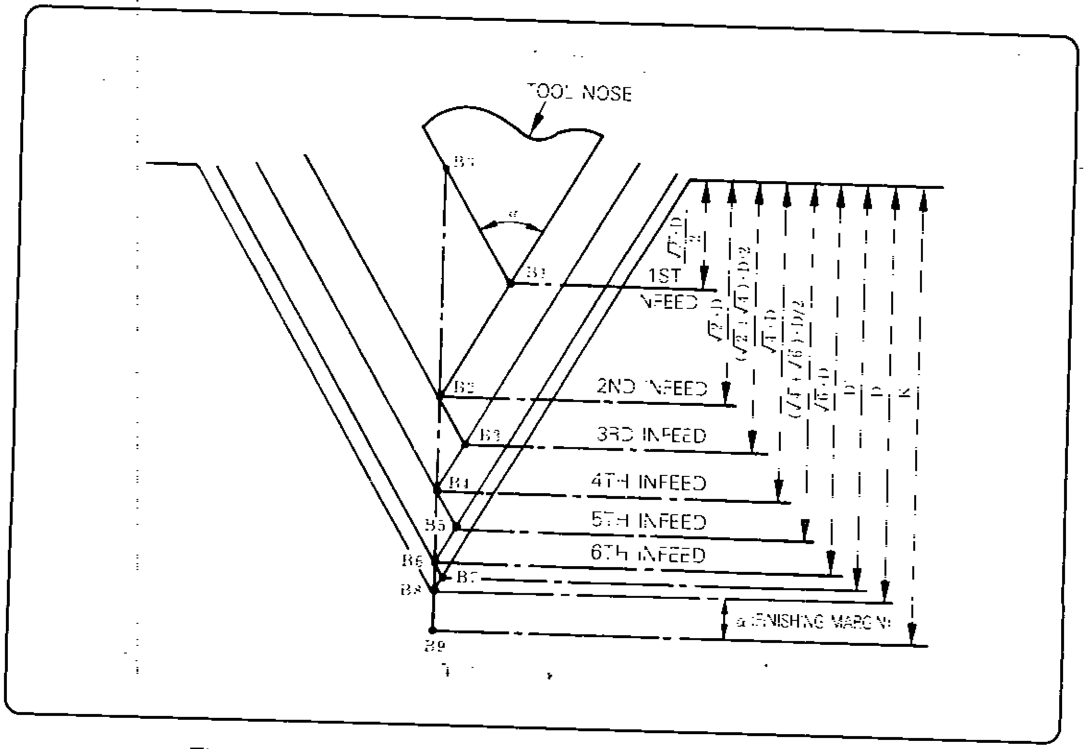


Fig. 2.13.45 Constant Cutting Amount, Zig-zag Cutting

NOTES ON G76

1. When taper thread cutting is commanded by A ≠ 0

When taper thread cutting is commanded by A ≠ 0, the X-axis coordinate at the thread cutting start position is not exactly the infeed depth Dn.

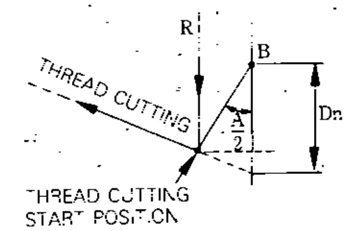


Fig. 2.13.46 Start Position when Taper Thread Cutting is Commanded by A ≠ 0

2. When unacceptable thread angle is commnded

When thread angle other than the 6 types (0°, 29°, 30°, 55°, 60°, 80°) is commanded, execution is performed with the next larger acceptable angle.
(Example) When commanded A15, execution is performed with A29; when A > 80 deg, execution is performed with A80.

3. When $\sqrt{nend}$ D does not match with (K-a)

When the final infeed depth $\sqrt{nend}$ D along the thread angle does not match with (K-a), the difference is subtracted from the 1st infeed amount D. The 1st infeed amount is never larger than D.

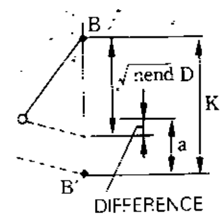


Fig. 2.13.47 1st Infeed when $\sqrt{nend}$ D ≠ (K-a)

(Example) When D = 5.0mm,
K = 9.8mm, constant amount a = 0.2mm

$$\begin{cases} \sqrt{nend}\ D = \sqrt{4} \times 5.000 = 10.000\text{mm} \\ \text{Difference} = \sqrt{nend}\ D - (K - a) \\ \quad = 10.000 - (9.800 - 0.200) = 0.400 > 0 \end{cases}$$

Therefore, each infeed depth is given as follows.
1st infeed ... 5.000 − 0.400 = 4.600mm
2nd infeed ... $\sqrt{2}$ × 5.000 − 0.400 = 6.671mm
3rd infeed ... $\sqrt{3}$ × 5.000 − 0.400 = 8.261mm
4th infeed ... $\sqrt{4}$ × 5.000 − 0.400 = 9.600mm
5th infeed ... 9.600 + 0.200 = 9.800mm
(a)

NOTES ON G76 (Cont'd)

4. Theread cutting feed hold (option)

When equipped with this option, the FEED HOLD button can be depressed to stop the thread cutting and return to start point A.

When parameter pm4011 D2 = 1, it can be made to stop at the position where the thread cutting is stopped. The cycle start button can then be depressed to return to point A.

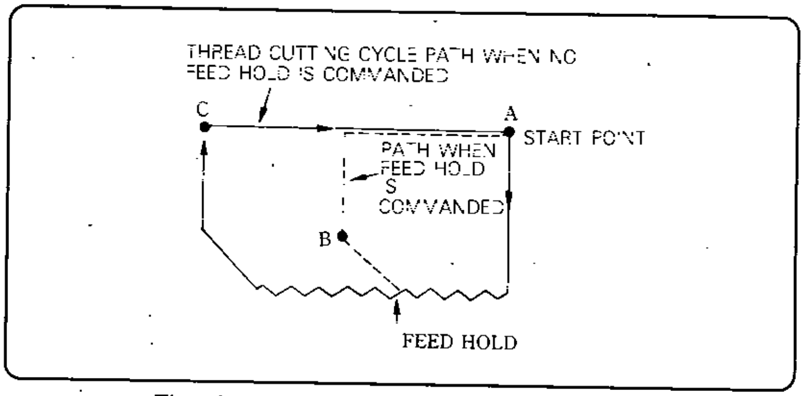


Fig. 2.13.48 Thread Cutting Feed Hold

When the thread cutting feed hold option is not provided, thread cutting continues when the FEED HOLD button is depressed, and does not stop until it completes the clearance at point C.

6. Nose R is disabled against the G76 cycle

7. The group 01 G code must be specified newly in the block after the G76 cycle.

   (Example) G76.......... ;
   <u>G00</u> M30 ;

2.13.2.9 Remarks on G70 to G76

(1) MDI mode is disabled

(a) Operation under the MDI mode is disabled during execution of the multiple repetitive canned cycle.

(b) G70 to G76 cannot be executed through operation under the MDI mode.

(2) Single-block operation

When the G70 to G76 cycles are executed with the single-block switch on, operation is performed as shown in Table 2.13.9.

Table 2.13.9 Single-block Operation

G70, G71, G72, G73, G74, G75	Stops at each minimum-unit block.
G76	 Stops at point A after each cycle.

(3) Symmetrical cutting pattern

Four symmetrical cutting patterns can be commanded by the G71 to G76 cycle commands (Fig. 2.13.49).

Command G71 to G73 with the finishing form program direction against point A.

Four patterns can be commanded by the G74 to G76 command positions (X, Z) or (U, W) against point A.

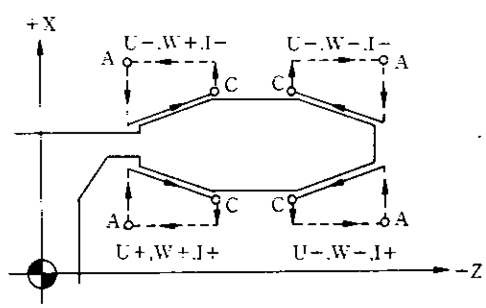


Fig. 2.13.49 Four Patterns

Table 2.13.10 List of Multiple Repetitive Cycles

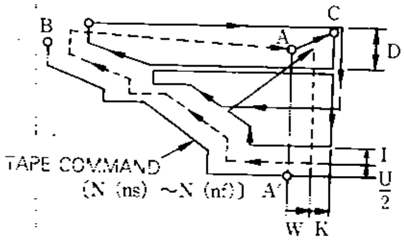
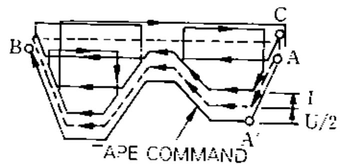
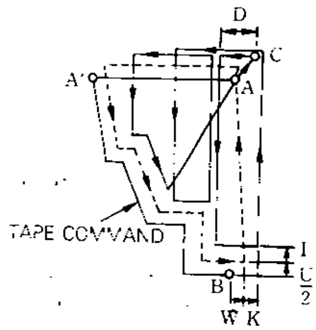
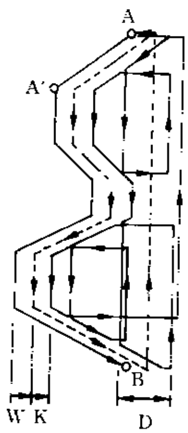
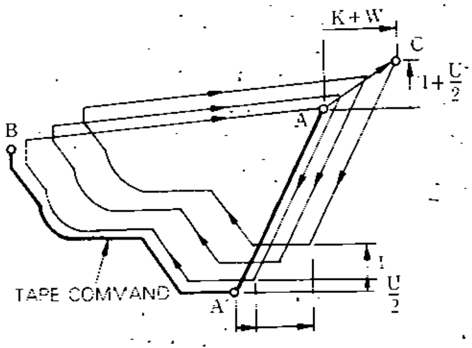
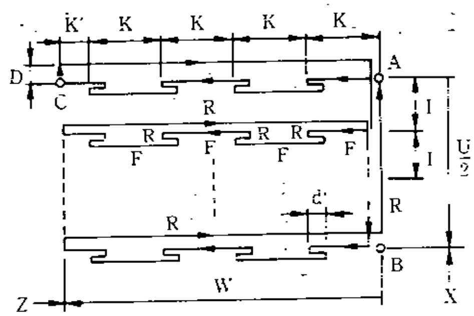
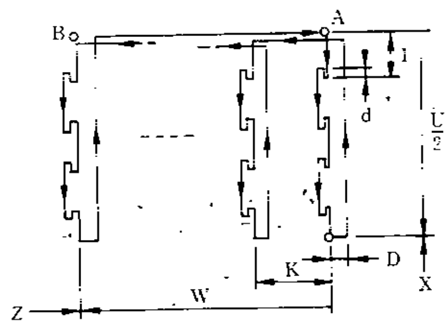
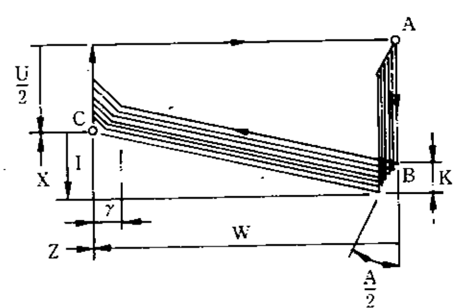
	Cutting Cycle	Command
G71 Outer Diameter Rough Cutting	(1)  (2)  	(1) Simple increase/decrease form G71 Pns Quf U.....W.....I.....K..... D.....F(E).....S..... ; Nns....... ; ⋮ } Finishing form Nnf....... ; (2) Form with dent G71 Pns Quf U.....I.....D.....F(E)..... S.....<u>R1</u> ; Nns....... ; ⋮ } Finishing form Nnf....... ; (U, W, I, K are designated with sign)
G72 End Face Rough Cutting	(1)  (2)  	(1) Simple increase/decrease form G72 Pns Quf U.....W.....I.....K..... D.....F(E).....S..... ; Nns....... ; ⋮ } Finishing form Nnf....... ; (2) Form with dent G72 Pns Quf W.....K.....D.....F(E)..... S.....<u>R1</u> ; Nns....... ; ⋮ } Finishing form Nnf....... ; (U, W, I, K are designated with sign)

Table 2.13.10 List of Multiple Repetitive Cycles (Cont'd)

	Cutting Cycle	Command
G73 Closed Loop Cutting		G73 Pns Qnf U.....W.....I.....K.....D..... F(E).....S..... ; Nns....... ; ⋮ } Finishing form Nnf....... ; (U, W, I, K are designated with sign)
G70	Executes Nns to Nnf finishing cutting	G70 Pns Qnf ;
G74 End Face Cutting Off		G74 } X(U).....Z(W)..... G75 I.....K.....D.....F(E).....R1 ; (1) Without "R1" command Moves as shown in the left figure. (2) With "R1" command Disregards the return amount d for each infeed, and returns to the point A level each time. d: Setting pm0864 (G74) pm0865 (G75) (I, D, K are designated without sign) (3) The infeed step can be commanded rather than the infeed amount, by address A. (4) The tool width can be commanded for shift equivalent to the tool width at the beginning and end, by address B. (I, K, D, A, B are designated without sign)
G75 Outer Diameter Cutting Off		
G76 Automatic Thread Cutting		G76 X(U).....Z(W).....I.....K.....D..... F(E).....A..... ; A: Thread angle (0°, 29°, 30°, 55°, 60°, 80°) (K, D are designated without sign 1/6 K≦D≦K)

2.13.3 Complex Chamfering (G111, G112) *

This command is used for tapering the workpiece and chamfering, rounding the arc section.

Table 2.13.11 G111, G112 Codes

G Code	Details
G111	Taper complex chamfering, rounding command
G112	Arc complex chamfering, rounding command

This command can be used to perform chamfering and rounding without the complex calculation at the taper and arc sections.

2.13.3.1 Taper complex chamfering, rounding command (G111) *

Taper→Chamfering, rounding→taper→chamfering, rounding

The above four moves are commanded in a single block

Fig. 2.13.50 shows typical forms commanded by the taper complex chamfering, rounding function.

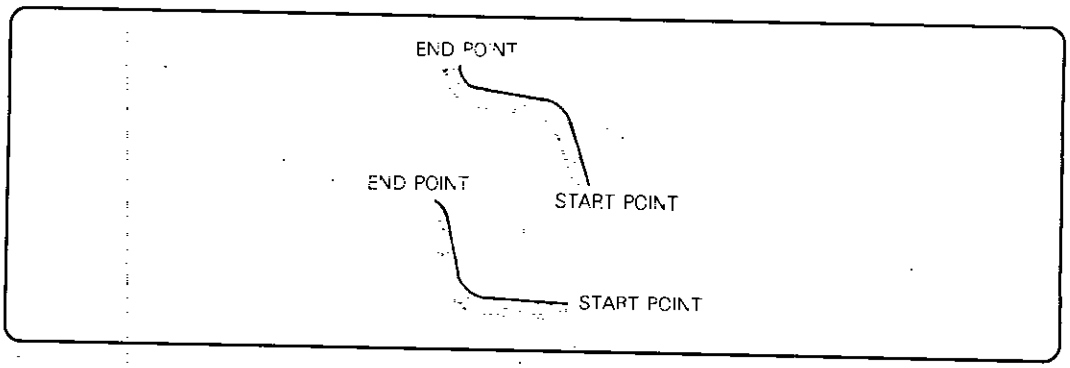


Fig. 2.13.50 Typical Forms of Taper Complex Chamfering, Rounding

(1) Various forms and command formats :

```
G111 X(U)······ I ······A······B······P······Q······ ;
G111 X(U)······ K ······A······B······P······Q······ ;
G111 X(U)······ I ······K······B······P······Q······ ;
```

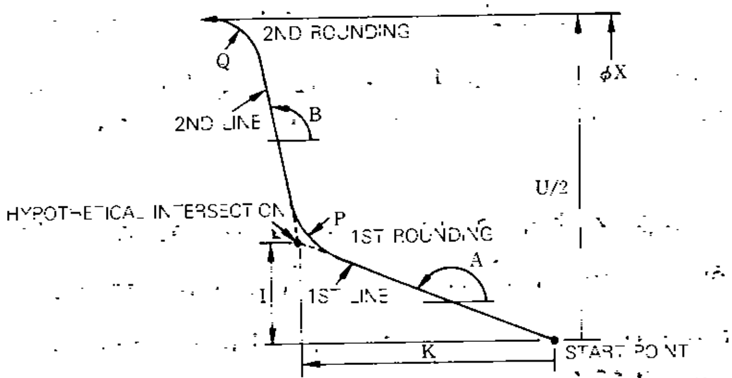


Fig. 2.13.51 Form and Command Format of Taper, Complex Chamfering, Rounding 1

```
G111 Z(W)······ I ······A······B······P······Q······ ;
G111 Z(W)······ K ······A······B······P······Q······ ;
G111 Z(W)······ I ······K······B······P······Q······ ;
```

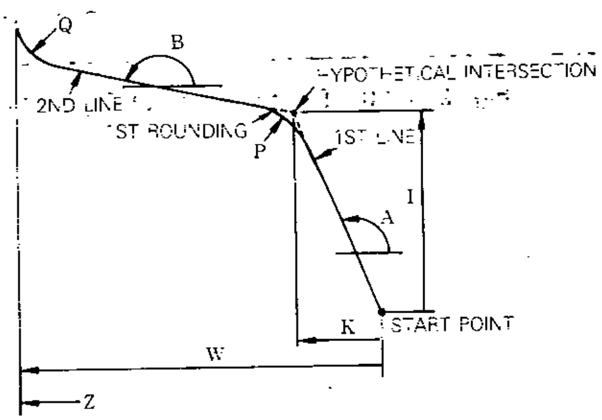


Fig. 2.13.52 Form and Command Format of Taper, Complex Chamfering, Rounding 2

```
G111 X(U)······ I ······A······B······C······D······ ;
G111 X(U)······ K ······A······B······C······D······ ;
G111 X(U)······ I ······K······B······C······D······ ;
```

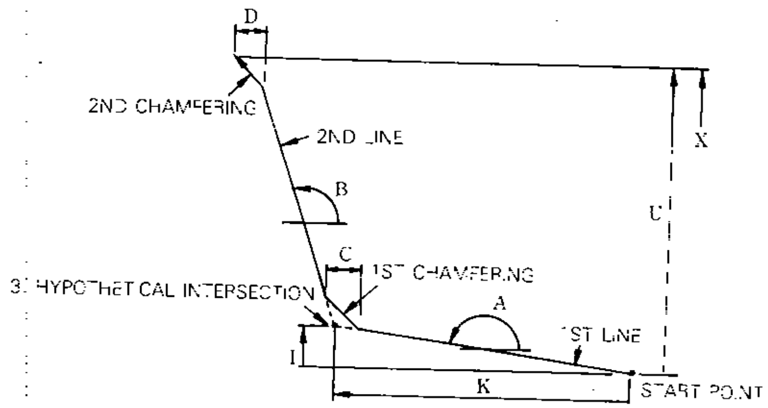


Fig. 2.13.53 Form and Command Format of Taper, Complex Chamfering, Rounding 3

```
G111 Z(W)······ I ······A······B······C······D······ ;
G111 Z(W)······ K ······A······B······C······D······ ;
G111 Z(W)······ I ······K······B······C······D······ ;
```

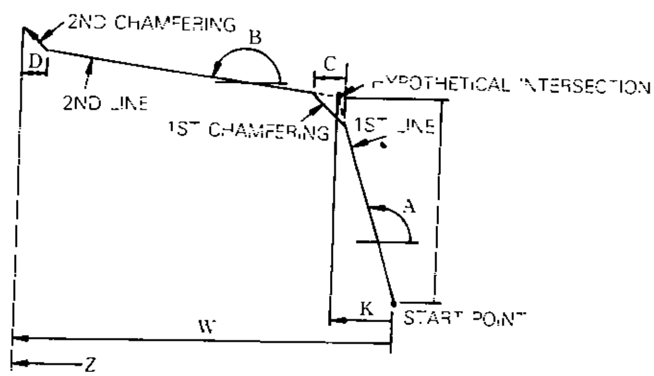


Fig. 2.13.54 Form and Command Format of Taper, Complex Chamfering, Rounding 4

(2) Address

The address works as shown in the following table can be specified in taper complex chamfering, rounding command.

Table 2.13.12 Address Words

Address Word	Description	Setting Unit
X(U)	X-axis end point coordinate (U: Incremental amount from start point)	1 = 0.001 mm or 1 = 0.0001 in
Z(W)	Z-axis end point coordinate (W: Incremental amount from start point)	
A	1st line move angle	1 = 0.001 deg
B	2nd line move angle	
I	X distance from hypothetical intersection of 1st and 2nd lines, and start point (radius value)	1 = 0.001 mm or 1 = 0.0001 in
K	Z distance from hypothetical intersection of 1st and 2nd lines, and start point	
P	1st rounding radius (without sign)	
Q	2nd rounding radius (without sign)	
C	1st chamfering amount (without sign)	
D	2nd chamfering amount (without sign)	

(3) Form designation

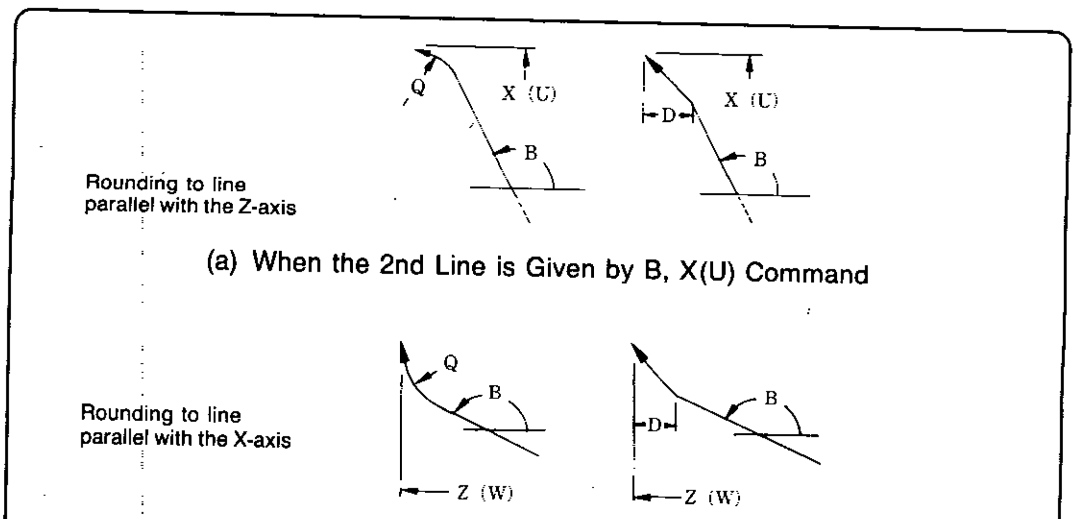
(a) Command the form according to **Table 2.13.13**.

Table 2.13.13 Form Designation

Form	Designation	
1st Line.	A: 1st line move angle I: X-axis distance from start point to hypothetical intersection K: A-axis distance from start point to hypothetical intersection	Command two of the three.
1st Chamfering or Rounding	C: 1st chamfering P: 1st rounding radius	Specify either.
2nd Line	B: Move angle from 2nd line X(U): X-axis end point coordinate U: X-axis end point Incremental value from start point Z(W): Z-axis end point W: Z-axis end point Incremental value from start point	Command two from the three. Note that X command and U command, or Z command and W command cannot be commanded together.
2nd Chamfering or Rounding	D: 2nd chamfering amount Q: 2nd rounding radius	Specify either.

(b) The 1st rounding is the rounding of the 1st and 2nd lines.

(c) The 2nd chamfering, rounding is commanded as shown is **Fig. 2.13.55** by the 2nd line command.



(a) When the 2nd Line is Given by B, X(U) Command

(b) When the 2nd Line is Given by B, Z(W) Command

Fig. 2.13.55 2nd Chamfering, Rounding

(d) The 2nd chamfering, rounding direction is commanded to the forward direction of the 2nd line.

Details are given in **Tables 2.13.14** and **2.13.15**.

Table 2.13.14 Direction of 2nd Chamfering

2nd Line Move Angle B Command Value	Direction of Chamfering (+x, −z)	Other Conditions
B=0, −360.000, 360.000	X, Z positive direction chamfering	When the 1st line is directed to the X positive direction
	X negative, Z positive direction chamfering	When the 1st line is directed to the X negative direction
0<B<90.000 −360.000<B<−270.000	X, Z positive direction chamfering	—
B=90.000, −270.000	X, Z positive direction chamfering	When the 1st line is directed to the Z positive direction
	X positive, Z negative direction chamfering	When the 1st line is directed to the Z negative direction
90.000<B<180.000 −270.000<B<−180.000	X positive, Z negative direction chamfering	—
B=180.000, −180.000	X positive, Z negative direction chamfering	When the 1st line is directed to the X positive direction
	X negative, Z negative direction chamfering	When the 1st line is directed to the X negative direction
180.000<B<270.000 −180.000<B<−90.000	X, Z positive direction chamfering	—
B=270.000, −90.000	X, Z negative direction chamfering	When the 1st line is directed to the Z negative direction
	X negative, Z positive direction chamfering	When the 1st line is directed to the Z positive direction
270.000≤B<360.000 −90.000<B<0	X negative, Z positive direction chamfering	—

Table 2.13.15 Direction of 2nd Rounding

2nd Line Move Angle B Command Value	Rounding +x −z	Other Conditions	
B=0, 360.000, −360.000	X negative, Z positive direction rounding	When the 1st line is directed to the X negative direction	X(U) command cannot be used
	X, Z positive direction rounding	When the 1st line is directed to the X positive direction	
0<B<90.000, −360.000<B<−270.000	X, Z positive direction rounding Rounding the line parallel to the X-axis	When the 2nd line is commanded by B, Z(W) command	
	Rounding the line parallel to the Z-axis	When the 2nd line is commanded by B, X(U) command	
B=90.000, −270.000	X, Z positive direction rounding	When the 1st line is directed to the Z positive direction	Z(W) command cannot be used
	X positive, Z negative direction rounding	When the 1st line is directed to the Z negative direction	
90.000<B<180.000, −270.000<B<−180.000	X positive, Z negative direction rounding Rounding the line parallel to the Z-axis	When the 2nd line is commanded by B, X(U) command	
	Rounding the line parallel to the X-axis	When the 2nd line is commanded by B, Z(W) command	
B=180.000, −180.000	X positive, Z negative direction rounding	When the 1st line is directed to the X positive direction	X(U) command cannot be used
	X, Z negative direction rounding	When the 1st line is directed to the X negative direction	
180.000<B<270.000, −180.000<B<−90.000	X, Z negative direction rounding Rounding the line parallel to the X-axis	When the 2nd line is commanded by B, Z(W) command	
	Rounding the line parallel to the X-axis	When the 2nd line is commanded by B, X(U) command	
B=270.000, −90.000	X, Z negative direction rounding	When the 1st line is directed to the Z negative direction	Z(W) command cannot be used
	X negative, Z positive direction rounding	When the 1st line is directed to the Z positive direction	
270.000<B<360.000, −90.000<B<0	X negative, Z positive direction rounding Rounding the line parallel to the Z-axis	When the 2nd line is commanded by B, X(U) command	
	Rounding the line parallel to the X-axis	When the 2nd line is commanded by B, Z(W) command	

(e) Supplementary explanation

When B, X(U), and Z(W) of the 2nd line are all commanded

When B, X(U), and Z(W) of the 2nd line are all commanded, one of the 1st line A, I, K can also be commanded:

Taper complex chamfering, rounding command

Command 1st line, 2nd line with address X, Z, I, K, A, B in the taper complex chamfering, rounding command. Not commanding address X, Z, I, K, A, B is different from commanding "0", so unlike other G commands, the "0" command cannot be committed.

Table 2.13.16 Omitting Address "0" Command in the Taper Complex Chamfering, Rounding Command

Address	Omitting the "0" Command
X Z I K A B	"0" command cannot be omitted
P Q C D	"0" command can be omitted (The chamfering, rounding amount becomes "0".)

When the 2nd line specified with the X(U), Z(W) commands

When the 2nd line is specified with the X(U), Z(W) commands, no 2nd chamfering rounding can be done, or error will occur.

Combination of chamfering

Combination of 1st chamfering, 2nd rounding, or 1st rounding, 2nd chamfering is also available.

When 1st line addresses A, I, K are all specified

When 1st line addresses A, I, K are all specified, A is ignored, and the I, K commands are effective on the 1st line.

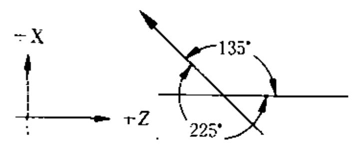
Ignore address B

When the 2nd line addresses B, X(U), Z(W) are all specified, and two of the 1st line addresses A, I, K are specified, B is ignored, and the 2nd line is formed by X(U), Z(W).

Commanding the line move angles A, B

Command positive value for the counter clockwise rotary angle of the Z-axis positive direction, and negative value for the clockwise direction rotary angle.

(Command value $-360.000 \leqq A, B \leqq 360.000$)



A135. or A-225.

NOTES ON G111

1. **G111 is non-modal G code**

   G111 is a non-modal G code, and is effective only within the commanded block.

2. **Commanding M, S, T**

   Address M, S, T cannot be commanded in the G111 block.

3. **Address C**

   Address C cannot be commanded in the G111 block if the 1st chamfering section is as shown in **Fig. 2.13.56**, **Fig. 2.13.57**.

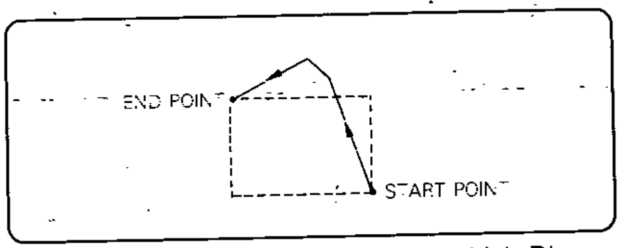


Fig. 2.13.56 Outside the Rectangle which Diagonal Line is the Start and End Points

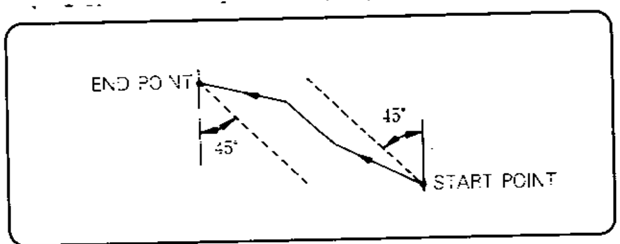


Fig. 2.13.57 Between the 45 Deg Line from the Start Point to the End Point and the 45 Deg Line from the End Point to the Start Point

   The end point is the end point of the 2nd line when there is no 2nd chamfering, rounding.

4. **When the G111 block is executed by the single block command**

   When the G111 block is to be executed by the single block command, the move to the end point must be commanded within 4 blocks.

5. **G111 block command**

   The NC performs all the operation for the 1st, 2nd line, 1st, 2nd chamfering/rounding, as soon as the G111 block is read in the buffer.

   The operation time can be more than 500 ms according to the form. If the operation time is longer than the move time of the previous block, the move stops, to cause inferior effect to the cutting surface.

   It is recommended to command and buffer (M93 command) the G111 block some blocks prior to the move, to prevent the move to stop by long operation time.

NOTES ON G111 (Cont'd)

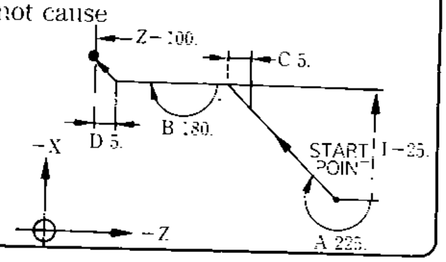
6. Notes on 1st line, 2nd line command

Error may or may not occur "when the 1st line is a 45-degree line directed from the start point to the end point" or "when the 2nd line is a 45-degree line directed from the end point to the start point".

(Example 1)

```
G111  Z-100. I-25. A225.
      B180. C5. D5. ;
```

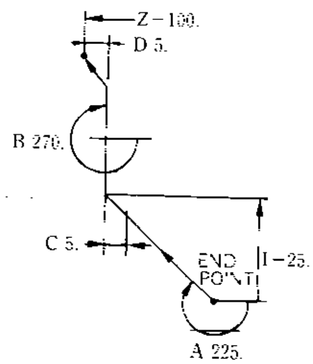
The above command does not cause error.



(Example 2)

```
G111  Z-100. I-25. A225.
      B270. C5. D5. ;
```

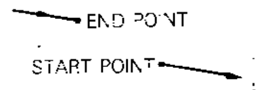
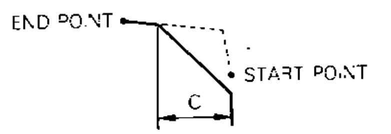
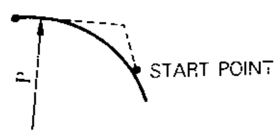
The above command causes error



NOTES ON G111 (Cont'd)

7. List of alarms caused by illegal G111 command

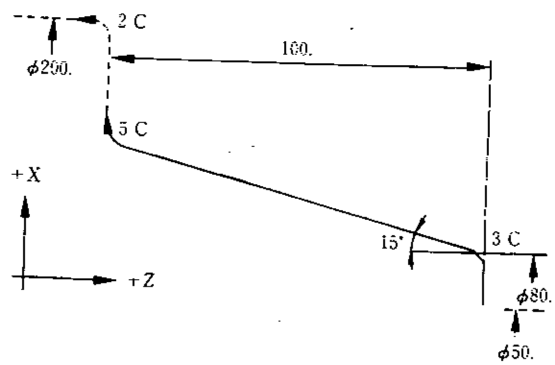
Table 2.13.17 Alarms Caused by Illegal G111 Command

<table>
<tr><th>Alarm Code</th><th>Description</th></tr>
<tr><td>0281</td><td>Only one of the 2nd line designation addresses B. X(U). Z(W) is commanded</td></tr>
<tr><td>0281</td><td>Two of the 2nd line designation addresses B. X(U). Z(W) are commanded, with only one or no 1st line designation address A. I. K specified.</td></tr>
<tr><td>0281</td><td>Both the 1st chamfering designation address C and 1st rounding designation address P are commanded.</td></tr>
<tr><td>0281</td><td>Both the 2nd chamfering designation address D and 2nd rounding designation address Q are commanded.</td></tr>
<tr><td>0282</td><td>The linear move angle designation address A. B command value is out of the −360.000≦A. B≦360.000 range.</td></tr>
<tr><td>0283</td><td>The 1st chamfering section is out of the rectangle including the start point and the end point.</td></tr>
<tr><td>0283</td><td>1st chamfering section is between the 45 deg line from the start point to the end point and the 45 deg line from the end point to the start point</td></tr>
<tr><td>0284</td><td>When there is no intersection of the 1st and 2nd lines

</td></tr>
<tr><td>0284</td><td>The 1st line and the 2nd line are on the same line.</td></tr>
<tr><td>0285</td><td>M. S. T are commanded in the G111 command block.</td></tr>
<tr><td>0284</td><td>The 1st line designation addresses A. I. K command values are as follows. and the form cannot be decided.
<table><tr><th>A Command Value</th><th>Command</th></tr><tr><td>−360.000. −180.000
0. 180.000. 360.000</td><td>Address I is commanded to the 1st line command</td></tr><tr><td>−270.000. −90.000
90.000. 270.000</td><td>Address K is commanded to the 1st line command</td></tr></table></td></tr>
<tr><td>0284</td><td>The 2nd line designation addresses B. X(U). Z(W) command values are as follows. and the form cannot be decided.
<table><tr><th>B Command Value</th><th>Command</th></tr><tr><td>−360.000. −180.000
0. 180.000. 360.000</td><td>Address X(U) is commanded to the 2nd line command</td></tr><tr><td>−270.000. −90.000
90.000. 270.000</td><td>Address Z(W) is commanded to the 2nd line command</td></tr></table></td></tr>
<tr><td>0284</td><td>The chamfering designation addresses C. D command values are too large against the commanded form that it cannot be operated as commanded.

</td></tr>
<tr><td>0284</td><td>The rounding radius designation addresses P. Q command values are too large against the commanded form that it cannot be operated as commanded.

</td></tr>
<tr><td>0281</td><td>The 2nd line is address X. Z command. and 2nd chamfering. rounding Q. D are commanded.</td></tr>
</table>

Example of Programming

(G01 W..... ;) → The 50 dia. (dotted line) command in the following figure

G111 W-100. I15. A90. B165. C3. D5. ; } Command of the following solid line part
G111 W-100. I15. K0. B165. C3. D5. ; }



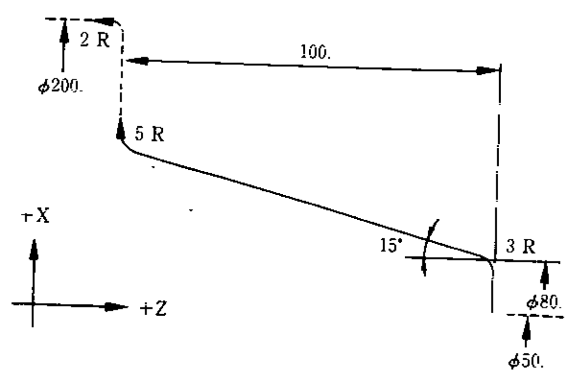
(a) Taper Complex Chamfering

⋮

(G01 W......;)→The 50 dia. (dotted line) command in the following figure

G111 W-100. I15. A90. B165. P3. Q5. ; } Command of the following solid line part
G111 W-100. I15. K0. B165. P3. Q5. ; }

(G12 X200. K-2. ;)←Command of the dotted line after solid line



(b) Taper Complex Rounding

2.13.3.2 Circular complex chanfering, rounding command (G112) *

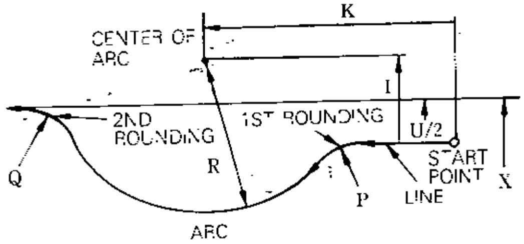
Line→Chamfering, rounding→arc→chamfering, rounding

The above four moves are commanded in a signal block.

(1) Various forms and command formats

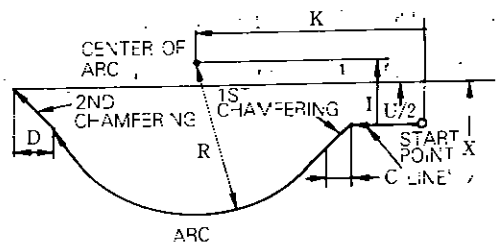
(a) Diameter side arc complex rounding

G112 X(U)······I······K······P······Q······R······ ;



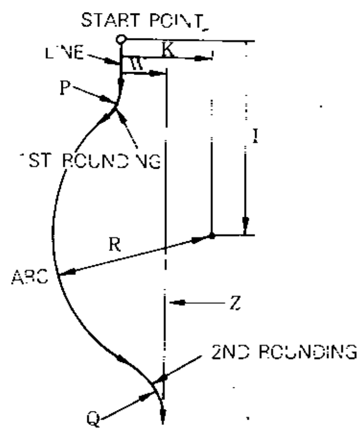
(b) Diameter side arc complex chamfering

G112 X(U)······I······K······C······D······R······ ;



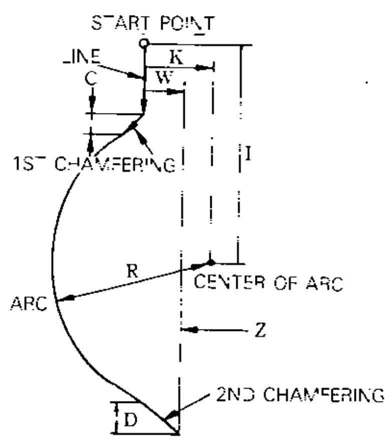
(c). Front arc complex rounding

G112 Z(W)……I……K……P……Q……R…… ;



(d) Front arc complex chamfering

G112 Z(W)……I……K……C……D……R…… ;



(2) Address

The address words as shown in the following **Table 2.13.18** can be specified in arc complex chamfering, rounding command.

Table 2.13.18 Address Words

Address Word	Description	Setting Unit
X(U)	X-axis end point coordinate (U: Incremental amount from start point) of diameter side arc complex chamfering, rounding command	1 = 0.001 mm or 1 = 0.0001 in (decimal point input enabled)
Z(W)	Z-axis end point coordinate (W : Incremental amount from start point) of front arc complex chamfering, rounding command	
I	X distance from arc center start point	
K	Z distance from arc center start point	
R	Arc radius	
P	1st rounding radius (without sign)	
Q	2nd rounding radius (without sign)	
C	1st chamfering amount (without sign)	
D	2nd chamfering amount (without sign)	

(3) Form designation

(a) Arc complex chamfering, rounding form

Table 2.13.19 shows the forms of the arc complex chamfering, rounding parts.

Table 2.13.19 Form

Form	Description
Line	Line form the start point that is parallel with the Z-axis (diameter side arc) or X-axis (front arc).
Arc	Arc which center is specified by I, K and is drawn from the start point
1st Chamfering	Chamfering done at the intersection of a line and arc, for the dimension specified by the C command.
1st Rounding	Rounding of line and arc specified by the radius of the P command.
2nd Chamfering	Chamfering performed by the D command at the intersection of arc and line based on the X(U) command that is parallel with the Z-axis (diameter side arc) or line based on the Z(W) command and prallel with the X-axis (front arc).
2nd Rounding	Rounding performed by the radius specified by the Q command that contacts with the arc and line based on the X(U) command that is parallel with the Z-axis (diameter side arc) or line based-on the Z(W) command-and parallel with the X-axis (front arc).

(b) Arc cutting direction

The arc cutting rotary direction is decided as shown in **Fig. 2.13.58**, so that the arc is at the opposite side of the arc center against the line drawn from the start point.

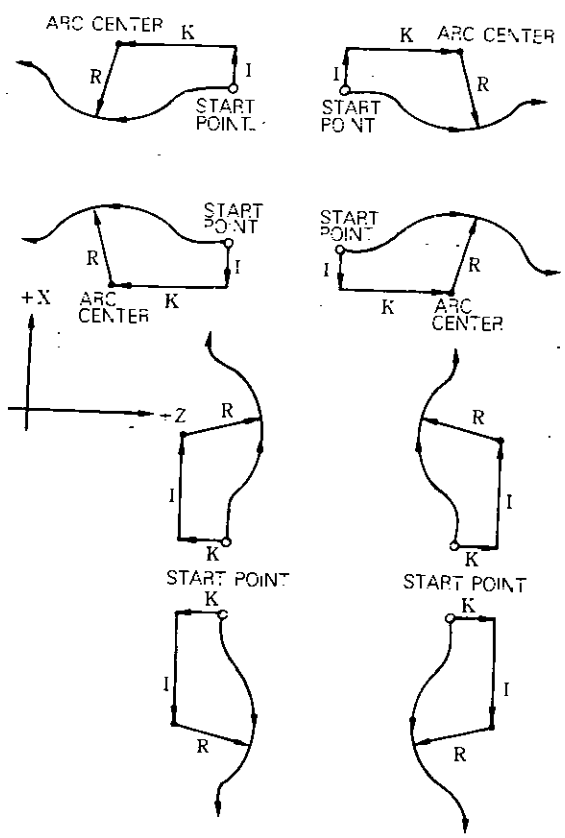


Fig. 2.13.58 Arc Cutting Rotary Direction

(c) Judging the arc cutting direction

The above explanation may not be sufficient in some command forms. Table 2.13.20 describes details on how the arc cutting direction is decided within the control unit.

Table 2.13.20 I, K Commands and Arc Rotary Direction

I, K Command Value	Arc Rotary Direction	
	Diameter Side Arc	Front Arc
I≧0, K≧0	Counterclockwise: CCW (G03 equivalent)	Clockwise: CW (G02 equivalent)
I≧0, K<0	Clockwise: CW (G02 equivalent)	Cunterclockwise: CCW (G03 equivalent)
I<0, K≧0	Clockwise: CW (G02 equivalent)	Cunterclockwise: CCW (G03 equivalent)
I<0, K<0	Counterclockwise: CCW (G03 equivalent)	Clockwise: CW (G02 equivalent)

The above arc direction can be reversed by specifying negative value for the arc radius R, as shown in Fig. 2.13.59.

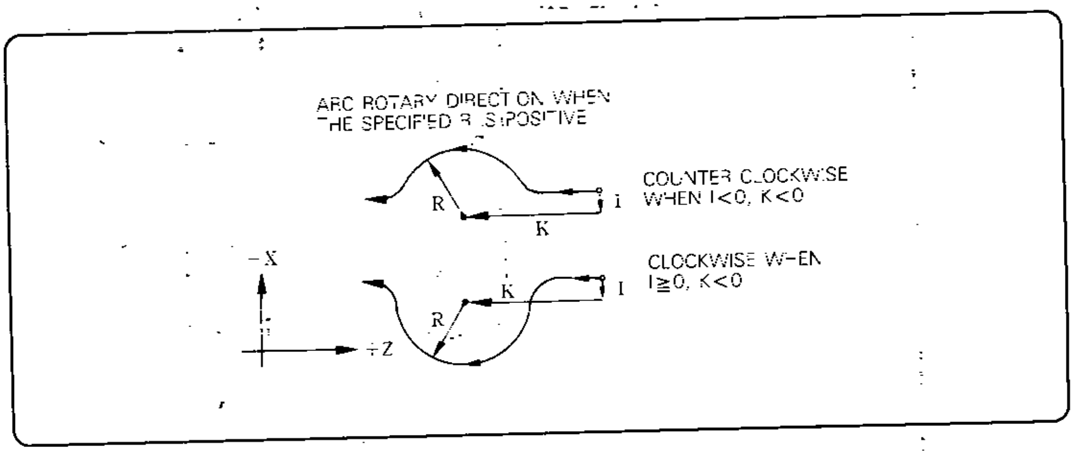


Fig. 2.13.59 Rotary Direction of Arc when the Specified R is Negative

(d) Omitting the address word

Address words X(U), Z(W) decides whether the arc is diameter side arc or front arc. The addresses cannot be omitted even when the end point and the start point are of the same coordinate (always command "U0" or "W0"). **Table 2.13.21** shows what happens if other address words are omitted.

Table 2.13.21 Omitting the Address Word

Address Word	When Omitted
I	Accepted as "I0" command
K	Accepted as "K0" command
R	Causes "R0" command alarm
P Q C D	Accepted as "0" command and no chamfering nor rounding is performed.

NOTES ON G112

1. G112 is non-modal G code

G112 is a non-modal G code, and is effective only within the commanded block.

2. Commanding addresses M, S, T

Address M, S, T cannot be commanded in the G112 block.

3. Single block command

When the G112 block is to be executed by the single block command, the move to the end point must be performed within 4 blocks.

4. Command in G71, G72, G73

When G112 command is used in the finishing form block of multiple repetitive cycle G71 (outer diameter rough cutting cycle), G72 (end face rough cutting cycle), or G73 (pattern shift cycle), the G112 command block is equivalent to 5 blocks.

5. G code command other than the G112 command

Do not specify other G codes in the G112 block, or it will cause alarm.

6. Preventing the operation time to stop the move

The NC performs all the operation for the line, arc, 1st, 2nd chamfering/rounding, as soon as the G112 block is read into the buffer.

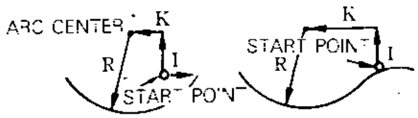
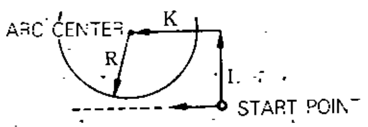
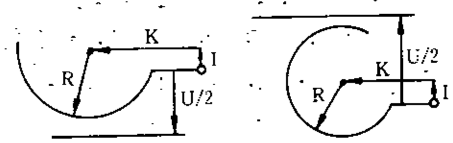
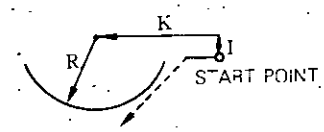
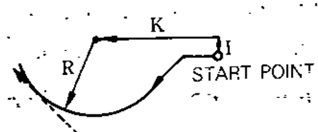
The operation time can be more than 500 ms according to the form. If the operation time is longer than the move time of the previous block, the move stops, to cause inferior effect to the cutting surface.

It is recommended to command and buffer (M93 command) the G112 block some blocks prior to the move, to prevent the move to stop by long operation time.

NOTES ON G112 (Cont'd)

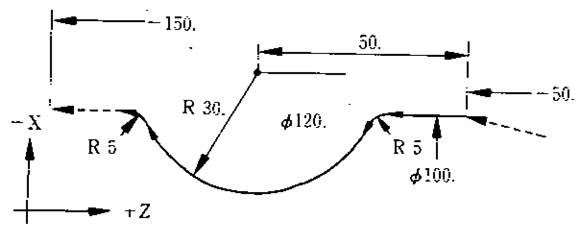
7. List of alarms caused by illegal G112 command

Table 2.13.22 Alarms Caused by Illegal G112 Command

Alarm Code	Description
0286	X(U) or Z(W) is not commanded.
0286	Both X(U) and Z(W) are commanded.
0286	R is not commanded. Or "0" is commanded.
0286	I, K is not commanded. Or "0" is commanded to both.
0286	P, C are both commanded
0286	Q, D are both commanded
0285	M, S, T is commanded
0287	Moves from the start point to the opposite direction from the arc center direction 
0287	The arc and line have no intersections. 
0287	The arc and end point command have no intersections. 
0287	The C command chamfering cannot be done. 
0287	The D command chamfering cannot be done. 

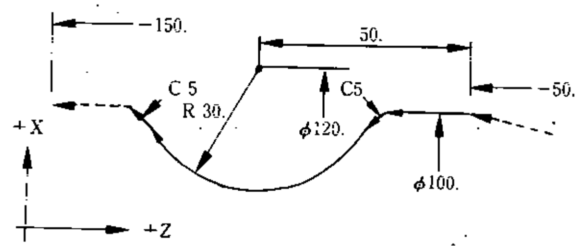
Example of Programming

```
(G01  X100.  Z-50. ;)←The dotted line before arc
 G112  U0  I10.  K-50.  P5.  Q5.  R30. ;
(G01  Z-150. ;)←The dotted line after arc
```



(a) Arc complex rounding

```
(G01  X100.  Z-50. ;)←The dotted line before arc
 G112  U0  I10.  K-50.  C5.  D5.  R30. ;
(G01  Z-150. ;)←The dotted line after arc
```



(b) Arc complex chamfering

2.13.4 Tool offset amount setting (G10) *

The tool offset amount can be set or corrected by the G10 command.

```
G10 P.....X(U).....Z(W).....R.....C..... ;
```

The above command allows setting or correction of the tool offset amount in the part program.

Table 2.13.23 Addresses

	Description
P	Specifies the tool offset number
X Z	Overwrites the tool offset to the specified amount
U W	Added the tool offset amount to the originally specified value
R	Overwrites the nose R amount to the specified amount
C	Tool nose control point data

The offset amounts of omitted addresses remain as is.

Example of Programming

```
G10  P16  X32.5  W0.05 ;
     |    |      |
     |    |      └── 0.005mm is added to the Z-axis
     |    └── 32.5mm is set to the X-axis
     └── Changes the tool offset number to "16"
```

(1) Tape format

The offset amount tape can be created in the above format and stored in the offset memory.

```
Label
%:
G10 P...X...Z...R... ;
G10 P...X...Z...R... ;
G10 P...X...Z...R... ;
%
```

(2) Work coordinate system shift memory data setting

`G10 P00 X(U).....Z(W).....C(H)..... ;`

The above command allows setting or correction of the work coordinate system shift data from the part program.

The following data is set in the work coordinate system.

X, Z, C :	Work coordinate system shift amount absolute setting data
U, W, H:	Work coordinate system shift amount incremental setting data

The shift amount is not changed if the address is omitted.

2

2.13.5 Program Call Function

2.13.5.1 Subprogram (M98, M99)

The subprogram assigned with program number and stored in the memory can be called and executed as desired.

M codes in **Table 2.13.24** are used.

Table 2.13.24 Subprogram

M Code	Description
M98	Subprogram call
M99	End of subprogram

(1) Calling the subprogram (M98)

M98 P.....Q.....L..... ;

The above command calls the subprogram with the program number pseciﬁed by P, that starts from sequence Q in the part program and executes the program for L times.

When P is omitted :	Calls subprogram that starts at sequence number Q in the main program
When Q is omitted:	Calls subprogram that starts from the head of the program number specified by P.
When L is omitted :	Executes the subprogram a single time.

Up to 4 nesting of subprogram call is allowed.

(2) End of subprogram (M99)

M99 ;

The above ends the subprogram.

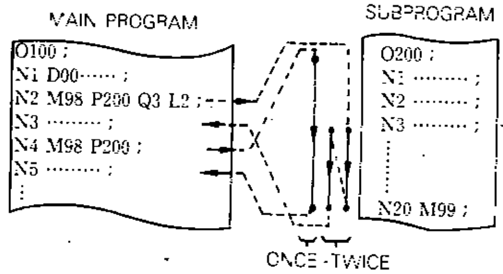
After execution of the subprogram, it returns to the next block in the main program.

M99 P....... ;

When the above is commanded at the end of the subprogram, it returns to the sequence number of the main program specified by P.

The following is an example of a program where the subprogram is called from the main program and executed.

Example of Programming



Subprogram Execution Sequence

(3) Simple jump instruction

M99 P........ ;

When the above is used in the main program, it simply jumps to the sequence number in the main program specified by P.

When P is omitted, it simply jumps to the head of the main program.

NOTES ON M98, M99

1. **Alarm "0262"**

   Alarm "0262" occurs when the program number specified by address P and the sequence number specified by address Q cannot be found.

2. **Conditions for simple jump**

   These functions can be used when the subprogram is stored in the part program memory. The main program can be commanded form the NC tape or part program memory.

3. **Alarm "0261"**

   Alarm "0261" occurs when the nesting of the subprogram exceeds 4 nestings.

4. **When M99; is commanded in the main program**

   When M99; is commanded in the main program, it returns to the head of the main program to perform endless operation.

2.13.6 Drilling Canned Cycle *

2.13.6.1 Drilling canned cycle *

(G80 to G89, G831, G841, G861) are programs that simplifies special drilling operations that consists for several blocks, into a single block command. Fourteen types of canned cycles are offered, and G80 is used to cancel the canned cycles.

(1) Canned cycle G code and operation

Table 2.13.25 shows the canned cycle G code and operation.

Table 2.13.25 List of Drilling Canned Cycle

G Code	Cutting In	At bottom	Clearance	Application
G80			—	Cancel
G81	Cutting feed	—	Rapid feed	Drill
G82	Cutting feed	Dwell	Rapid feed	Boring
G83	Intermittent feed	—	Rapid feed	Deep hole drilling
G831	Intermittent feed	—	Rapid feed	High speed deep hole drilling
G84	Cutting feed	Reverse spindle after dwell	Cutting feed→dwell →forward spindle	Tapping
G841	Cutting feed	Reverse spindle after dwell	Cutting feed→dwell →reverse spindle	Reverse tapping
G85	Cutting feed	—	Cutting feed	Boring
G86	Cutting feed	Spindle stop	Rapid feed→spindle forward	Boring
G861	Cutting feed	Spindle indexing →shift	Rapid feed→shift Spindle forward	Boring
G87	Spindle indexing→shift →rapid feed→shift spindle→spindle forward→cutting feed	Spindle indexing →shift	Rapid feed→shift→ spindle forward	Back boring
G88	Cutting feed	Spindle stop after dwell	Manual return→ spindle forward	Boring
G89	Cutting feed	Dwell feed	Cutting feed	Boring

(2) Command format

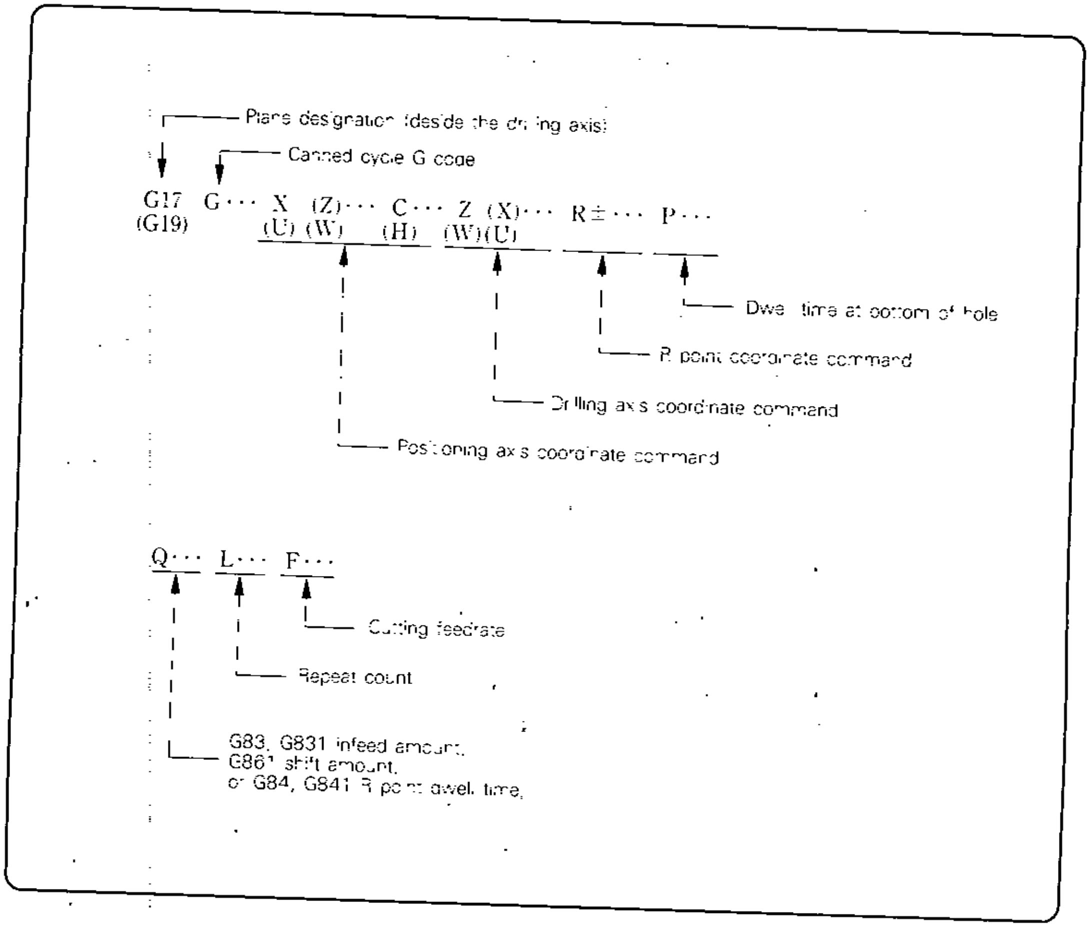


Fig. 2.13.60 Drilling Canned Cycle

The command of **Fig. 2.13.60** executes the following ① to ④ as a single cycle.

① Positioning at the drilling position
② Rapid feed to the R point
③ Drilling to the bottom
④ Return to the R point or initial point

(a) Initial point

The initial point is the absolute position of the drilling axis when the mode shifted from the canned cycled cancel mode to canned cycle mode.

The initial point does not changed by executing canned cycle under the G199 (R point level return) mode.

(b) Address

- Positioning axis: Command the drilling position by incremental or absolute value.
- Drilling axis : Commnad the position of the bottom of the hole by absolute value of incremental value from the R point. The movement from the R point to the bottom of hole is done by G01 with the speed specified by the F code.

  The G00 movement may be included in some canned cycles (intermittent feed, etc.).

  Return from the bottom of hole to the R point is done by G00 or G01, according to the canned cycle.

Table 2.13.26 Plane Designation G code and Drilling Axis

G Code	Designated Plane (Positioning Plane)	Drilling Axis
G17	XY	Z
G18	ZX	Y
G19	YZ	X

NOTES
1. The G18 plane is usually, in 2-axis (X, Z) lathe, for normal cutting.
   Always command G17 or G19 when starting the drilling canned cycle, and G18 when canceling the drilling cycle.
2. The C-axis is the positioning axis, whatever the designation of G17 to G19.
3. Alarm occurs when the plane designation does not coincide with the plane designation by setting pm4017 D6=1 in the drilling canned cycle.

- R : Command the R point position in absolute value or incremental value from the initial point. The operation axis is the drilling axis. Return from the R point to the initial point is done by G00. In standard G code, the R point command is always absolute. When the drilling axis is the X-axis, the command must be given in the same unit as the X-axis (by diameter value when parameter pm1000 D1 = 0, and radius value when parameter pm1000 D1 = 1).
- L : Command the repeat count by address L. When this L is omitted, the count is regarded to be 1.
- P : Command the dwell time at bottom of hole. The unit is 1 = 1 ms, and 1 second when P1.0 is commanded. When P is omitted, no dwell is done.
- Q : Command the G83, G81 infeed amount and G861, G87 shift amount.

  Command in incremental value without sign, and radius value for X-axis component.

(3) Return mode designation

The return mode when ending one cycle of the drilling canned cycle differs according to the G code.

G198—Returns to the initial point
G199—Returns to the R point

The above G codes are modal.

(4) Cycle operation

Table 2.13.27 Drilling Canned Cycles

- Dwell
- Single block stop

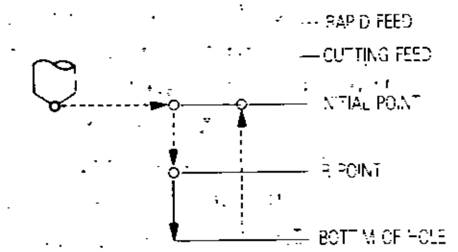
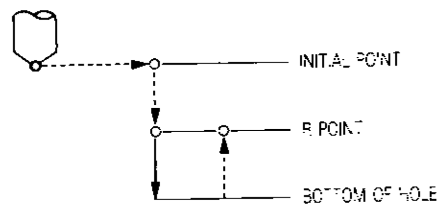
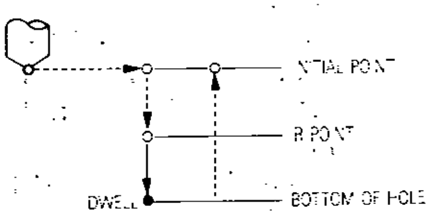
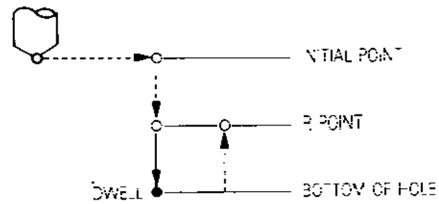
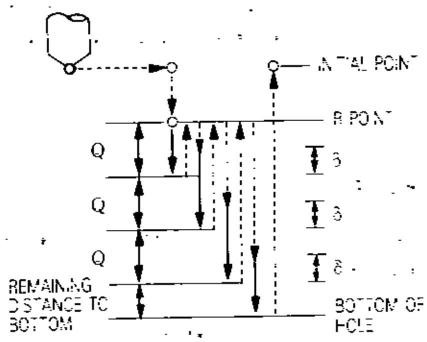
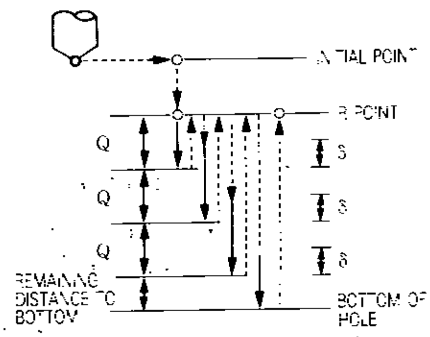
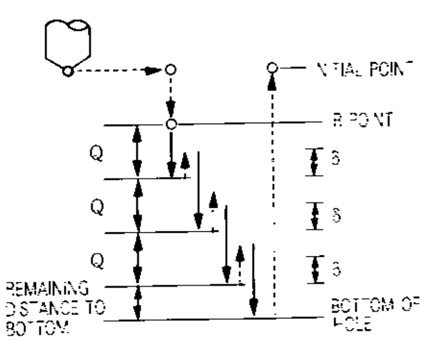
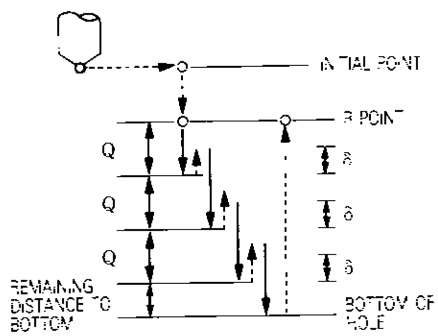
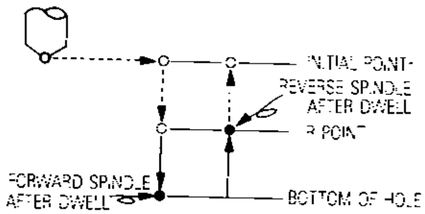
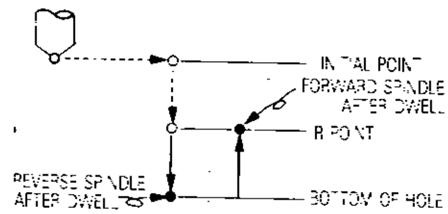
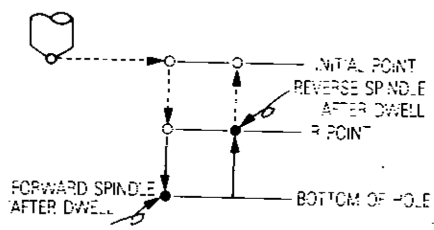
<table>
<tr><th></th><th>G198 (Initial Point Return)</th><th>G199 (R Point Return)</th></tr>
<tr><td>G81 Drilling</td><td></td><td></td></tr>
<tr><td>G82 Drill Boring</td><td></td><td></td></tr>
<tr><td>G83 Deep Hole Drilling</td><td>δ : SETTING (G17 : pm0870
G18 : pm0871
G19 : pm0872)</td><td>δ : SETTING (G17 : pm0870
G18 : pm0871
G19 : pm0872)</td></tr>
<tr><td>G831 High-speed Deep Hole Drilling</td><td>δ : SETTING (G17 : pm0870
G18 : pm0871
G19 : pm0872)</td><td>δ : SETTING (G17 : pm0870
G18 : pm0871
G19 : pm0872)</td></tr>
</table>

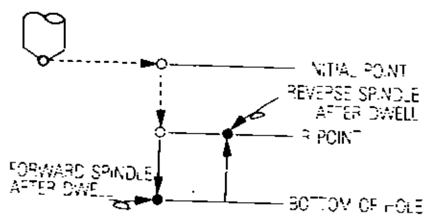
Table 2.13.27 Drilling Canned Cycles (Cont'd)

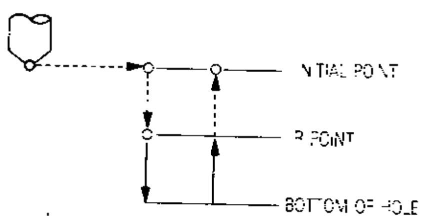
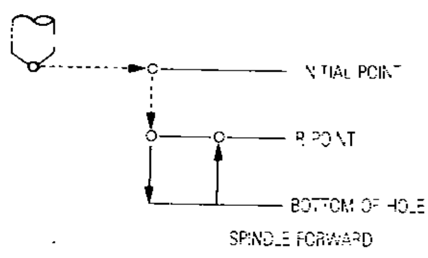
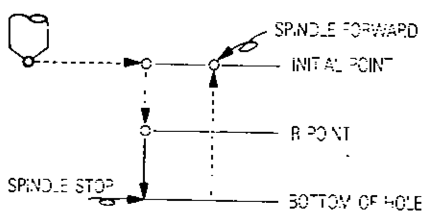
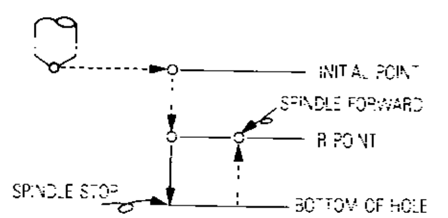
• Dwell
• Single block stop

<table>
<tr><th></th><th>G198 (Initial Point Return)</th><th>G199 (R Point Return)</th></tr>
<tr><td>G84 Tapping</td><td>

P : DWELL TIME AT BOTTOM OF HOLE
Q: DWELL TIME AT R POINT</td><td>

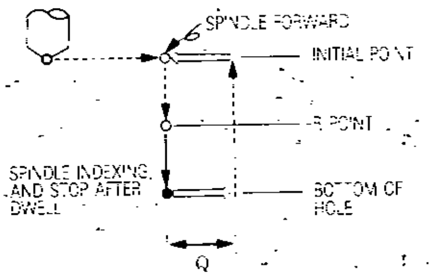
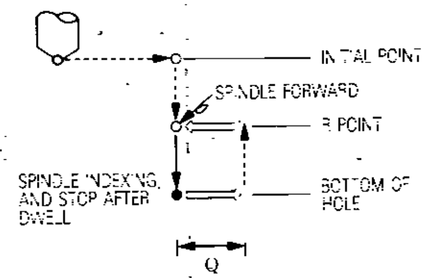
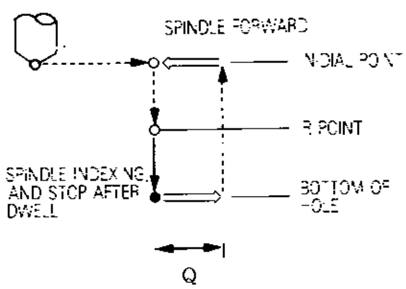
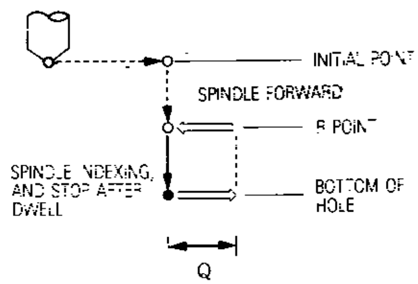
P : DWELL TIME AT BOTTOM OF HOLE
Q: DWELL TIME AT R POINT</td></tr>
<tr><td></td><td colspan="2">NOTE See descriptions on the spindle control of the drilling canned cycle (5) for spindel control.</td></tr>
<tr><td>G841 Reverse Tapping</td><td>

P : DWELL TIME AT BOTTOM OF HOLE
Q: DWELL TIME AT R POINT</td><td>

P : DWELL TIME AT BOTTOM OF HOLE
Q: DWELL TIME AT R POINT</td></tr>
<tr><td>G85 Boring</td><td>
</td><td>
</td></tr>
<tr><td>G86 Boring</td><td>
</td><td>
</td></tr>
</table>

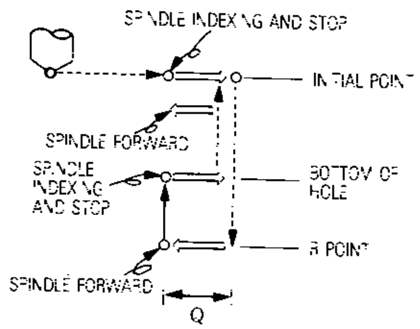
- Dwell
- Single block stop

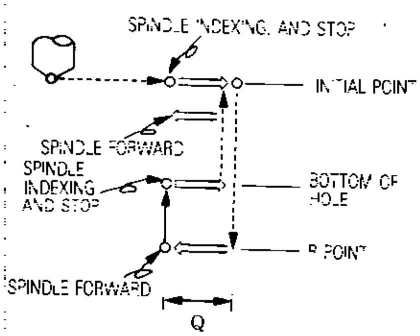
Table 2.13.27 : Drilling Canned cycles (Cont'd)

	G198 (Initial Point Return)	G199 (R Point Return)
G861 (Fixed Shift) Boring	 Q: Shift amount (incremental value without sign) Sift speed: pm2864 Shift direction: pm4028 Dwell time is commanded by P	 Q: Shift amount (incremental value without sign) Sift speed: pm2864 Shift direction: pm4028 Dwell time is commanded by P
G861 (Variable Shift) Boring	 Q: Shift amount (specified by i, j, k) G17: Specified by i, j G18: Specified by k, i G19: Specified by j, k i : X axis incremental value (with sign) (radius value) j : Y axis incremental value (with sign) k : Z axis incremental value (with sign) Shift speed: pm2864 Dwell time is commanded by P	 Q: Shift amount (specified by i, j, k) G17: Specified by i, j G18: Specified by k, i G19: Specified by j, k i : X axis incremental value (with sign) (radius value) j : Y axis incremental value (with sign) k : Z axis incremental value (with sign) Shift speed: pm2864 Dwell time is commanded by P

• Dwell
• Single block stop

Table 2.13.27 Drilling Canned cycles (Cont'd)

<table>
<tr><th></th><th>G198 (Initial Point Return)</th><th>G199 (R Point Return)</th></tr>
<tr><td>G87 (Fixed Shift) Back Boring</td><td>

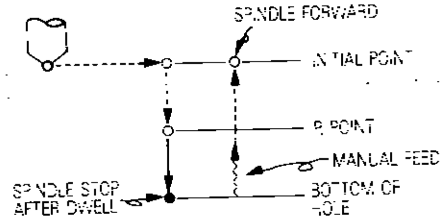
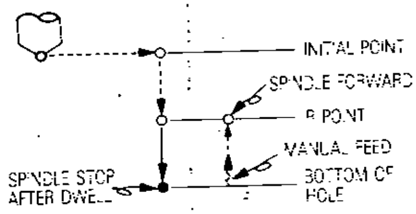
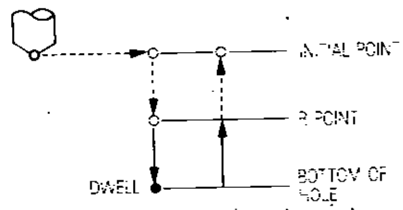
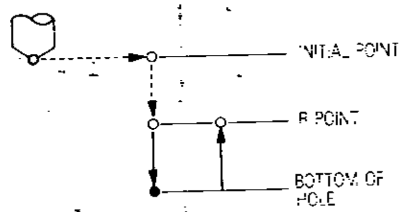
Q: Shift amount (incremental value without sign)
Sift speed: pm2864
Shift direction: pm4028</td><td>Not Used</td></tr>
<tr><td>G87 (Fixed Shift) Back Boring</td><td>

Q: Shift amount (specified by i, j, k)

G17: Specified by i, j
G18: Specified by k, i
G19: Specified by j, k
i : X axis incremental value (with sign) (radius value)
j : Y axis incremental value (with sign)
k : Z axis incremental value (with sign)
Shift speed: pm2864</td><td>Not Used</td></tr>
</table>

- Dwell
- Single block stop

Table 2.13.27 Drilling Canned Cycles (Cont'd)

	G198 (Initial Point Return)	G199 (R Point Return)
G88 Boring		
G89 Boring		

(5) Spindle control of drilling canned cycle

(a) G84, G841 tapping, reverse tapping cycles

Table 2.13.28 Spindle Control of Tapping, Reverse Tapping Cycles

	G84	G841
At Bottom of Hole	(Spindle stop) Spindle reverse	(Spindle stop) Spindle forward
At Clearance	(Spindle stop) Spindle forward	(Spindle stop) Spindle reverse

The following spindle control parameters can be set to output the value as M code.

Table 2.13.29 Spindle Control Parameters for Tapping, Reverse Tapping Cycles

	Line 1	Line 2	Line 3	No Parameter Setting
Spindle Forward	pm4430	pm4433	pm4436	M03
Spindle Reverse	pm4431	pm4434	pm4437	M04
Spindle Stop	pm4432	pm4435	pm4438	M05

When changing the rotary direction as spindle forward→reverse, reverse→forward, select the following parameter to stop the spindle once, or to directly change the direction.

pm4016 D3 0: Do not output the spindle stop M code
1: Output the spindle stop M code

(b) G86 boring cycle

Table 2.13.30 Spindle Control of G86 Boring Cycle

	G86
At Bottom of Hole	Spindle Stop
At Clearance	Spindle Forward

NOTE The spindle control M code parameter is the same as Table 2.13.29.

(c) G861, G87 boring, back boring cycles

Table 2.13.31 Spindle Control of G861, G87 Boring, Back Boring Cycles

	G861	G87
At Initial Point	—	Spindle indexing, stop
At R Point	—	Spindle forward
At Bottom of Hole	Spindle indexing, stop	Spindle indexing, stop
At Clearance	Spindle forward	Spindle forward

NOTE The spindle control M code parameter is the same as Table 2.13.29.

The following spindle indexing, stop parameters are used.

Table 2.13.32 Spindle Indexing Stop Parameters

	Line 1	Line 2	Line 3	No Parameter Setting
Spindle Indexing, Stop	pm4445	pm4446	pm4447	M19

(6) C-axis clamp/unclamp

The C-axis can be clamped during drilling. Set "1" in parameter #4017 D4, to output the C-axis clamp/unclamp M code as specified in the following table, at the position of the drilling cycle showm in Fig. 2.13.61. Dwell can also be inserted after unclamping the C-axis.

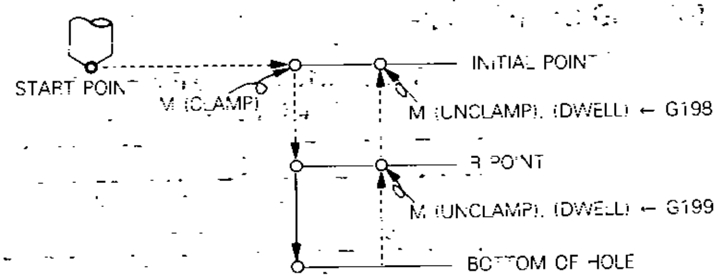


Fig. 2.13.61 C-axis Clamp/unclamp

Table 2.13.33 C-axis Clamp Parameters

	Line 1	Line 2	Line 3	No Parameter Setting
Clamp M Code	pm4439	pm4441	pm4443	No default. Always set the parameter
Unclamp M Code	pm4440	pm4442	pm4444	
Dwell Time after Clamp	pm0400			
Clamp/No Clamp	pm4017 D4	0: Clamp 1: No clamp		

(7) G861 (boring), G87 (back boring cycle) shift

(a) G861/G87 shift direction (by Q command)

The shift amount can be specified by the Q command, and the shift direction by parameter pm4028.

Table 2.13.34 Shift Direction and Parameter

Plane Designation	Drilling Axis	Shift Direction Parameter	Details
G17	Z-axis	pm4028 D0, D1	D1 D0 0 0 −X 0 1 −X 1 0 +Y 1 1 −Y
G18	Y-axis	pm4028 D3, D2	D3 D2 0 0 +Z 0 1 −Z 1 0 +X 1 1 −X
G19	X-axis	pm4028 D5, D4	D5 D4 0 0 +Y 0 1 −Y 1 0 +Z 1 1 −Z

(b) G861/G87 shift direction (by I, J, K)

Linear interpolation shift can also be commanded by I, J, K. Command the shift amount as follows, according to the plane.

G17 (XY plane): Command by I, J
G18 (ZX plane): Command by K, I
G19 (YZ plane): Command by J, K
i : X-axis incremental value (with sign) (radius value)
j : Y-axis incremental value (with sign)
k: Z-axis incremental value (with sign)

The shift speed is the value in parameter pm2863 for both (a) and (b).

```
(G17)  X...Z...R...I...
                   |
                   └── Shift amount command
```

NOTES

1. When addresses Q and I, J, K are commanded at the same time, addresses I, J, K are ignored.
2. I, J, K are modal in the drilling canned cycle.
3. Address Q is also in the drilling canned cycle, so if address Q is commanded before commanding the I, J, K commands, I, J, K are ignored.

   If I, J, K are needed, cancel the canned cycle once by G80, before commanding I, J, K.

(8) Modal status of drilling canned cycle command data

(a) G code

G81 to G89, G831, G841, G861 are modal till G code of the same group or G80, G code of group 01 is commanded.

(b) Drilling data (R point, drilling bottom point, Q, I, J, K, P)

The drilling data is modal in the drilling canned cycle.

NOTE Incremental command of the bottom point is counted from the R point. When the R point command alone is changed during the drilling canned cycle, the bottom coordinate will be changed to the incremental value from the new R point. Always command both the R point and bottom value to avoid such mistake.

(c) Repeat count L

The data of L is effective in the block only.

NOTES ON THE DRILLING CANNED CYCLE

1. When executing the drilling canned cycle with the single block function on

When the drilling canned cycle is executed with the single block switch on, execution stops in midstream and the feed hold lamp lights.

(a) After positioning at the positioning point
(b) After positioning at the R point
(c) After each cycle when L is commanded

The single block stop after ending the drilling canned cycle is done as usual, without lighting the single stop lamp.

2. R, hole bottom points when under the drilling canned cycle mode

When entering the drilling canned cycle mode, always command the R and bottom to decide the R point and bottom point. The R point and bottom point are erased when the drilling canned cycle is canceld.

3. When executing the drilling canned cycle changing the address data

When executing the drilling canned cycle changing the address data, address command is necessary in the block or in the next block. Without this command, the drilling canned cycle cannot be executed.

X(U), Z(W), C(H), Y(V), R command

4. When the M, S, T codes are commanded in the drilling canned cycle block

Commmanding the M, S, T codes in the drilling canned cycle blck causes the M, S, T to be sent at the first position (also when L is commanded).

M, S, T commands are generally not to be commanded alone.

NOTES ON THE DRILLING CANNED CYCLE (Cont'd)

5. **Alarm "170"**

   Alarm "170" occurs when the following G codes are commanded under the drilling canned cycle mode.

   - ✻ Group G codes except for G04
   - Nose R G codes (G41, G42)

   Therefore, always cancel the drilling canned cycle before commanding G50, G27, G28, etc.

6. **Single block insertion of dwell (G04)**

   A single block of dwell (G04) can be inserted in the middle of the drilling canned cycle mode. Dwell can be executed normally without trouble.

7. **Program alarm "182"**

   Program alarm "182" occurs when drilling canned cycle is commanded under the nose R (G41, G42) mode.

8. **Preparation before entering the drilling canned cycle mode**

   Before entering the drilling canned cycle mode, the spindle must always be started to the forward or reverse direction by automatic operation command. Switch between the forward/reverse rotations by manual operation, to be ready to enter the drilling canned cycle.

9. **Subprogram call under the drilling canned cycle (M98)**

   The subprogram can be called under the drilling canned cycle by commanding M98 P...L... ; The drilling canned cycle can also be continued by the called subprogram. The drilling canned cycle P designation (dwell time) is temporarily destroyed by the M98 P designation (jump destination program number), but the original value is returned after jumping to the subprogram. Functions as the maximum 4 nesting level limit and command function from the punch tape are the same as modes other than the drilling canned cycle mode. Alarm occurs when the drilling canned cycle and M98 are commanded in the same block.

10. **Drilling canned cycle repeat count L**

    The drilling canned cycle repeat count L is modal. However, the data is temporarily stored in the following special cases.

    (Example)

    G81 U10. R-20. Z-30. F100 ;

    L3 ;.... The drilling canned cycle is note executed because there is no X(U), Z(W), C(H), Y(V), R command.

    X20 L3 is stored.

    .... The stored L3 executes G82 3 times. L3 is then erased.

    As shown above, when L is specified in the drilling canned cycle, it is stored till it is actually executed.

11. **Canceling the drilling canned cycle**

    The drilling canned cycle is canceled when G80 or group 01 G code is commanded.

    Alarm "170" occurs when group 01 G code and drilling canned cycle G code are commanded in the same block. However, this does not apply when the drilling canned cycle G code is G80. (No alarm occurs, and the prescribed operation is executed.)

NOTES ON THE DRILLING CANNED CYCLE (Cont'd)

12. **Other modal G codes when the drilling canned cycle is command**

    When the drilling canned cycle is commanded after the drilling canned cycle cancel mode. the model G codes to the block immediately before are stored, and becomes effective when the drilling canned cycle is canceled.

13. **F commanded in the drilling canned cycle**

    The F commanded in the drilling canned cycle remains after the drilling canned cycle is canceled.

14. **Address search in the drilling canned cycle**

    Alarm occurs when address search is attempted in the middle of the drilling canned cycle (with block stop).. Address search can be done if the block stop is commanded by the program. When multiple cycles are executed in a single block (specified by L), alarm occurs if address search is attempted before the each cycle ends.

15. **When another drilling canned cycle is commanded during the drilling canned cycle**

    Another drilling canned cycle can be commanded during execution of a drilling canned cycle.

    If the address data is omitted in the new drilling canned cycle, the modal information from the previous blocks is applied.

2.14 MACRO PROGARM (G65, G66, G67) *

A group of instruction can be used to create program for a special task at the machine tool builder or user.

Such program is called the macro program. This program can be call and executed by 1-block command as G65 or G66.

The following can be done in the macro program.

① Variables can be used.
② Operation command using variables and constants can be given.
③ Control command for branch and repeat can be given.
④ Message and data can be output.
⑤ Arguments can be specified.

Therefore, complex programs with complex operations and conditions can be created without difficulty.

The following are the difference between the macro program and the sub-program.

DIFFERENCES
(1) Arguments can be specified in macro program call (G65, G66), but cannot be specified in subprogram call (M98).
(2) When command other than P, Q, L is found in the M98 block, it jumps to the subprogram only after executing the command. However, in G65, G66, commands other than P, L are regarded as arguments, so it jumps immediately to the macro program.
(3) The local variables used in the macro program correspond to the level of the macro program. The level of the local variables used in the subprogram does not change. In short, the local variables differs before and after the call in macro program, but remain the same in the subprogram.

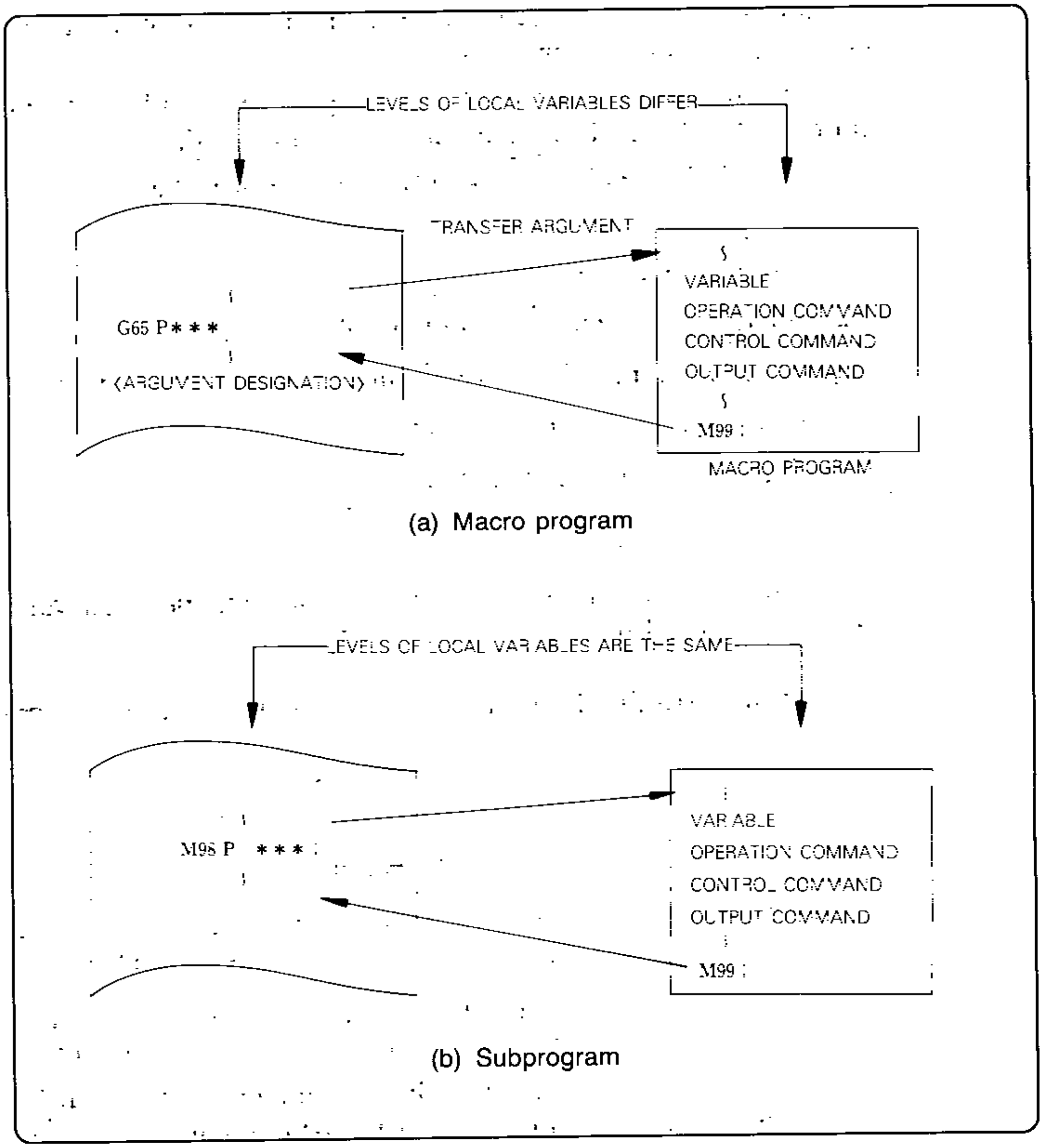


Fig. 2.14.1 Difference between Macro Program and Subprogram

2.14.1 Macro Program Call

The macro program is normally called and executed.

The macro program can be called as follows (Table 2.14.1).

Table 2.14.1 Macro Program Call

Call Method	Command Code	Remarks
Simple Call	G65	
Modal Call	G66	Canceled by G67
G code Call	G***	G command is 3 digits
M code Call	M***	M command is 2 digits, 3 digits
T code Call	T****	T command is 4 digits or 6 digits

(1) Simple call (G65)

G65 P.....L..... (Argument designation) ;

The macro program of the program number specified by P is called and executed L times by the above command.

The argument can be transferred to the macro program if necessary.

Table 2.14.2 Addresses

Address	Details	Digits
P	Program No.	5 digits
L	Repeat count	9 digits

Argument designation: Real number is assigned to the local variable that corresponds to the called macro program.
(See "argument designation" for details.)

NOTE Argument designation must be done before all the arguments of G65 are specified.

(2) Modal call (G66, G67)

The modal call command sets the mode to call the macro program. The macro program is actually called and executed only when certain conditions are satisfied.

| G66 P.....L..... (Argument designation) ; |

The mode to call the macro is set by the above command.

Then, the macro program with the program number specified by P is called and executed L times after the move commands in the block is completed, if any.

When argument is specified, the argument is transferred each time the macro program is called, as in the simple call command. The argument address and local variable correspond in the same way as in simple call (G65).

NOTE | G67 ; | cancels this mode.
When argument is to be specified. G66 must be commanded before all arguments.
When G66 is commanded, also command G67 is the same program.

(3) Macro program call by G code

| G✻✻✻ (argument designation) |

The above can be commanded to call and execute the macro program/subprogram of the program number corresponding to the G code.

Up to 24 sets of G✻✻✻ codes of maximum 3 digits can be set in the NC to call the macro program/subprpgram. Do not use existing G codes. Optional program numbers can be assigned to each set G✻✻✻.

Table 2.14.3 Setting Parameter

Sets	Called G code	Called Program Number
1	pm4480 max. 3 digits	pm4840 max. 5 digits
2 to 23	to	to
24	pm4503	pm4863

Max. 24 sets

When the system has multiple lines, the 24 sets of G codes in **Table 2.14.3** can be shared among the lines, or divided into two 12 sets (2 lines) or three 8 sets (3 lines) and used independently. The settingis done by parameter. ..... **i80L only**.

When parameter pm4019 D5 = 0, the G code sets are shared, and when it is D5 = 1, they are used independently.

When selected to be used independently, the G code sets are divided as follows.

- When there are 2 lines
  1st line pm4480 to pm4491 G codes
  2nd line pm4492 to pm4503 G codes
- When there are 3 lines
  1st line pm4480 to pm4487 G codes
  2nd line pm4488 to pm4595 G codes
  3rd line pm4496 to pm4503 G codes

(4) Macro program call by M code

M✻✻✻ (argument designation) ;

The above command calls and executes the macro program/subprogram of the program number corresponding to the M code.

Up to 24 sets of M✻✻✻ codes of maximum 3 digits can be set in the NC to call the macro program/subprogram. Do not use M00, M01, M02, M30 and internal processing M codes.

Optional program numbers can be assigned to each set M✻✻✻.

Table 2.14.4 Setting Parameter

Sets	Called M Code	Called Program Number
1	pm4504 max. 3 digits	pm4864 max. 5 digits
2 to 23	to	to
24	pm4527	pm4887

↑ Max. 24 sets

Parameter can be used to select whether the 24 pairs of **Table 2.14.4** are to be shared by the multiprogram systems or to be used independently by 12 pairs (2 programs), 8 pairs (3 programs). ..... **i80L only**

When parameter pm4019 D6 = 0, the 24 pairs are shared, when D6 = 1, they are used independently.

When specified for independent used, the number of programs decides the assignment.

- 2 programs (12 pairs each)
  1st program M code of pm4504 to pm4515
  2nd program M code of pm4516 to pm4527
- 3 programs (8 pairs each)
  1st program M code of pm4504 to pm4511
  2nd program M code of pm4512 to pm4519
  3rd program M code of pm4520 to pm4527

When multiple M codes are commanded in a single block, it checks whether the first M code is a macro call M code. The parameter regards the 2nd and after as normal M codes or argument designation.

pm4020 (D6) = 0: Normal M code
= 1: Argument designation

NOTE The MF M code is not output as when normal M code is commanded, when the marco program call M code is executed.

(5) Macro program call by T code

It can be selected whether the commanded T code is to be regarded as normal T command, or as macro call command. When the call program number (pm4889) is "0", the T code is the normal T command.

One pair of the program number to be called can be set optionally. When called, the T command value becomes the argument of common variable #149. No other argument can be specified.

Table 2.14.5 Settable Parameters

Command Selection	Call Program No.
Normal T Command	pm4889 = 0
Call T Command	pm4889 Max. 5 digits

In multi-program systems, is can be selected by parameter whether the pm4889 of **Table 2.14.5** is shared by the programs, or the T code macro is used independently by each program. .... **i80L only**

When parameter pm4019 D7 = 0, pm4889 is shared by each program; when D7 = 1, T code macro is used independently by each program.

The following marco call T commands are used when used independently.

- 2-program
  - 1st program pm4889
  - 2nd program pm4890
- 3-program
  - 1st program pm4889
  - 2nd program pm4890
  - 3rd program pm4891

NOTE T code TF is not output when executing the call T code. The T command is 4 digits or 6 digits.

NOTES ON G65, G66

The priority when the G, M, T macro program calls are commanded in the same block is:

G>M>T

(Example)

When commanded,

G...M...T... ;

G... alone is enabled, while M, T are disregarded. No alarm occurs.

T...M...G... ;

G... alone is enabled. regardless of the commanded sequence.

2.14.1.1 Macro nesting call

As subprograms can be called from subprograms, macro programs can also be called from macro programs.

The nesting is incremented when the macro program is called and executed once by the G65, G66, G, M, T codes.

The maximum nesting of the above macro calling codes is total 4 nestings.

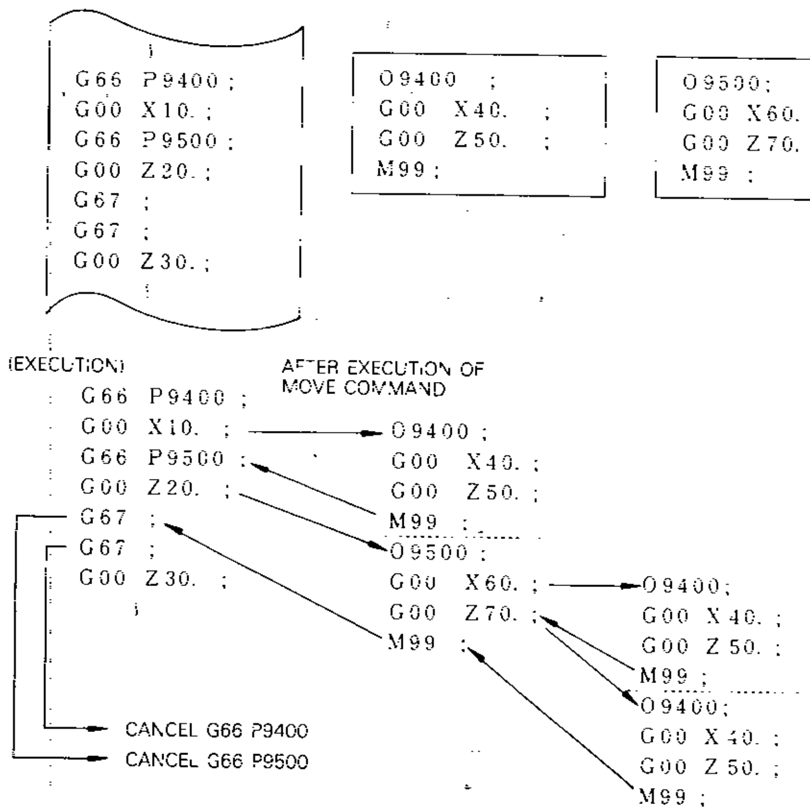
(1) Nesting of macro call by G, M, T codes

The nesting of macro call by the G, M, T codes is 1 nesting. In short, macro program call by the G, M, T codes cannot be done in a macro program call by the G, M, T codes.

If G, M, T is specified in the macro program called by the G, M, T codes, G code alarm occurs. The M, T codes are executed as ordinary M, T codes.

(2) Modal call G66

In modal call G66, the specified macro program is called and executed each time a move command is executed. When G66 is nested in the same program, it becomes effective against the move command in the called macro program, with priority on fist previous G66 command, then on the second previous G66 command ...... Execution is done from the macro program commanded later.



NOTE When G66 is nested, they must be canceled by commanding G67 from the later-commanded G66.

Fig. 2.14.2 Nested Macro Call

2.14.1.2 Argument designation

Argument designation is to assign real numbers to local variables used in the macro program. There are two types of argument designation; type I and type II. The two types can be mixed freely.

(1) Argument designation I addresses and local variables

Table 2.14.6 Argument Designation I Addresses and Local Variables

Cross Reference of Addresses and Variables	
Argument designation I address	Local variable
A	#1
B	#2
C	#3
D	#7
E	#8
F	#9
H	#11
I	#4
J	#5
K	#6
M	#13
Q	#17
R	#18
S	#19
T	#20
U	#21
V	#22
W	#23
X	#24
Y	#25
Z	#26

(2) Argument designation II addresses and local variables

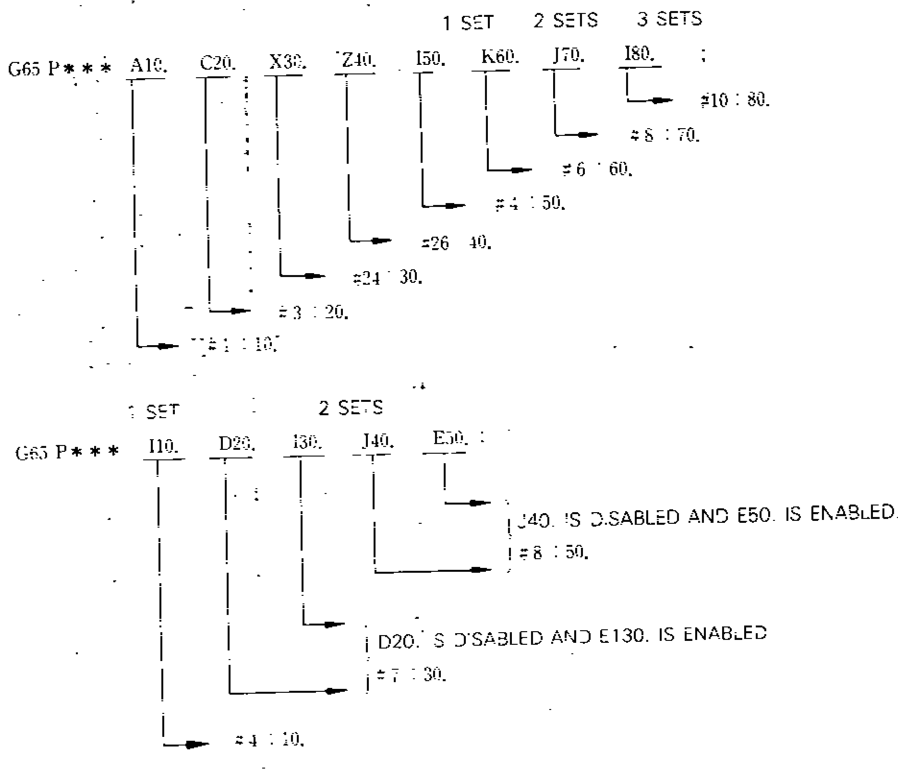
Table 2.14.7 Argument Designation II Addresses and Local Variables

Cross Reference of Addresses and Varibles	
Argument designation I address	Local variable
A	#1
B	#2
C	#3
I_1	#4
J_1	#5
K_1	#6
I_2	#7
J_2	#8
K_2	#9
I_3	#10
J_3	#11
K_3	#12
I_4	#13
J_4	#14
K_4	#15
I_5	#16
J_5	#17
K_5	#18
I_6	#19
J_6	#20
K_6	#21
I_7	#22
J_7	#23
K_7	#24
I_8	#25
J_8	#26
K_8	#27
I_9	#28
J_9	#29
K_9	#30
I_{10}	#31
J_{10}	#32
K_{10}	#33

NOTES

1. Always command I, J, K in this order if ever. In argument designation II, the suffix 1 to 10 of I, J, K is a set that indicates the sequence of the I, J, K sets, and is not written in actual command.
2. Argument designation can be omitted in address that does not need argument designation. The local variable that correspond to the address without argument designation just becomes (vacant).
3. When multiple addresses are commanded against a single variable number, the newly commanded address is accepted.
4. When multiple I, J, K are commanded, the sequence is decided with I, J, K as 1 set, so the variable numbers are decided according to this sequence.

- Example of argument designation



NOTE The macro program call code must always be specified before the argument designation.

Signs and decimal points can be used in the argument designation number regardless of the address.

If the decimal point is not applied, the numeral is stored in the variable as numeral with decimal point according to the normal address digit type.

Fig. 2.14.3 Example of Argument Designation

(3) Position of decimal point of argument

Arguments are generally specified with signs and decimal points. If no decimal point is specified, the positon of the decimal point is decided as shown in **Table 2.14.8**.

Table 2.14.8 Position of Argument Point

Argument Designation Address	Metric	Inch
A, B	3	3
D, H	0	0
E	4	6
F (Under G99 mode)	3	4
F (Under G98 mode)	3	4
I, J, K, C	3 (2)	4 (3)
M, S, T	0	0
Q	0	0
R	3 (2)	4 (3)
U, V, W	3 (2)	4 (3)
X, Y, Z	3 (2)	4 (3)

NOTES
1. The numeral indicates which digit from the right end the decimal point is to be.
2. ( ) are the digits below decimal point when parameter pm1000 D0=1.

2.14.2 Variables

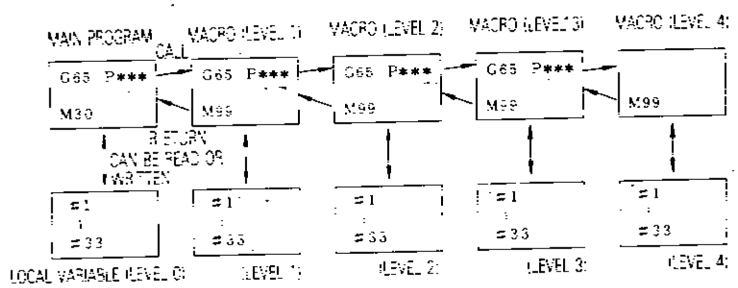
There are three types of variables: local variables, common variables, and system variables.

(1) Local variables (#1 to #33)

Local variables are locally used for each macro. Each time a macro is called, new local variables (#1 to #33) are secured for that macro. A local variable may take on a value of a specified argument, or it may assume the results of calculation in the macro.

Each time the macro is called, values of passed arguments are stored in local variables. Local variables to which no argument is passed become (null).

When a macro is returned by M99, the local variables for that macro become «null». Also, when the equipment is turned on or reset, all local variables become (null).



NOTES

1. In the main program, level-0 local variables are secured first. Other local variables are secured when a macro is called, corresponding to the levels (1-4) of the macro.
2. When a macro is called by G65, for example, the local variables at the level of the calling macro are saved, and new local variables are allocated at the level of the called macro. At that time, arguments can be passed between the macros.

   Consequently, a local variable of the same macro may have different values if the macro is called at different times.
3. When a macro returns to a macro of one level lower than that by instruction of M99, the local variables at the higher macro level are cleared to (null). The local variables that were saved when the macro was called are restored at the higher macro level.
4. MDI write of local variable:
   Local variables cannot be written from the operation panel. To assign a value to a local variable, specify the argument or write in the program.
5. Local variables can also be used in a subprogram. In a subprogram, the available local variables are at the current macro level. When a subprogram is called, arguments cannot be specified. When a subprogram is returned by M99, the local variables at the current macro level are not cleared to (null).

Fig. 2.14.4 Local Variables

(2) Common variables (#100 to #299, #500 to #999)

Common variables can be used in common by the main program, subprograms, and macros, or through a multi-level state of all of these. That is, a common variable that takes on the results of calculations made by a certain macro can be used in common by other macros. However, no argument can be specified for common variables.

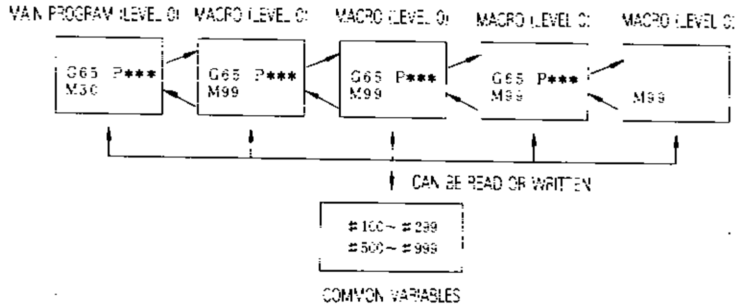


Fig. 2.14.5 Common Variables

(a) Common variables come in the following two types depending on the state when reset:

Table 2.14.9 Common Variables

#100 to #299	Variables become (null) when the equipment is turned on or reset. To prevent the variables from becoming (null) at reset, set parameter pm4009 D1 to 1.
#500 to #999	Values are retained when the equipment is turned on or reset.

(b) A number of sets of common variables can be added with option.

Table 2.14.10 Option Type and Number of Additional Sets

Option Type	Number of Sets			
	Common in program	Separate for 2 program	Separate for 3 program	
a	#100 to #149, #500 to #549	#100 to #124, #500 to #524	#100 to #115, #500 to #515	(For 1 program)
		#100 to #124, #500 to #524	#100 to #115, #500 to #515	(For 2 program)
			#100 to #115, #500 to #515	(For 3 program)
b	#100 to #199, #500 to #599	#100 to #149, #500 to #549	#100 to #132, #500 to #532	(For 1 program)
		#100 to #149, #500 to #549	#100 to #132, #500 to #532	(For 2 program)
			#100 to #132, #500 to #532	(For 3 program)
c	#100 to #199, #500 to #699	#100 to #149, #500 to #599	#100 to #132, #500 to #565	(For 1 program)
		#100 to #149, #500 to #599	#100 to #132, #500 to #565	(For 2 program)
			#100 to #132, #500 to #565	(For 3 program)
d	#100 to #299, #500 to #999	#100 to #199, #500 to #749	#100 to #165, #500 to #665	(For 1 program)
		#100 to #199, #500 to #749	#100 to #165, #500 to #665	(For 2 program)
			#100 to #165, #500 to #665	(For 3 program)

For multiple-program control, use parameter pm4020 D1 to specify whether to use the common variables separately for the sections.

If pm4020 D1 = 0, common variables are shared by programs. If pm4020 D1 = 1, different sets of common variables are used for different programs.

If different sets of common variables are allocated separately to programs, it is possible to share only #100 to #199 for the programs in common by setting parameter pm4020 D3 to 1.

(3) System variables

System variables are predetermined according to the application. There are the following types:

Table 2.14.11 System Variables

System Variable Type	System Variable No.
Interface input signal	#1000 to #1031, #1032 Note 1
Interface output signal	#1100 to #1131, #1132 Note 1
Tool offset, work coordinate system shift quantity	#2000 to #2499 #12001 to #13499 Note 2 #14101 to #14112
Alarm message display	#3000 Note 1
Clock	#3001, #3002, #3010 Note 3
Single-block stop and auxiliary function complete wait control	#3003 Note 1
Feed hold, feedrate override, exact stop control	#3004 Note 1
RS-232C data output	#3100
Modal information	#4000 to #4999 Note 1
Position information	#5000 to #5999 Note 1

NOTES
1. During multi-section control, these system variables are used in each program.
2. During multi-program control, these system variables are changed according to offset definitions. See (c), "Offset and work coordinate system shift quantity" below.
3. During multi-section control, part of these system variables are used in each program.

(a) Interface input signal

(i) To read the ON/OFF status of any of the 32 input signals used for a macro program, specify the corresponding system variable #1000 to #1031 on the right side of an expression. **Table 2.14.16** shows the correspondence between the input signals and the system variables:

Table 2.14.12 Interface Input Signals and System Variables

System Variable	#1007	#1006	#1005	#1004	#1003	#1002	#1001	#1000
Input signal	UI 7 2^7	UI 6 2^6	UI 5 2^5	UI 4 2^4	UI 3 2^3	UI 2 2^2	UI 1 2^1	UI 0 2^0
System variable	#1015	#1041	#1013	#1012	#1011	#1010	#1009	#1008
Input signa	UI 15 2^{15}	UI 14 2^{14}	UI 13 2^{13}	UI 12 2^{12}	UI 11 2^{11}	UI 10 2^{10}	UI 9 2^9	UI 8 2^8
System variable	#1023	#1022	#1021	#1020	#1019	#1018	#1017	#1016
Input signal	UI 23 2^{23}	UI 22 2^{22}	UI 21 2^{20}	UI 20 2^{20}	UI 19 2^{19}	UI 18 2^{18}	UI 17 2^{17}	UI 16 2^{16}
System variable	#1031	#1030	#1029	#1028	#1027	1026	#1025	#1024
Input signal	UI 31 2^{31}	UI 30 2^{30}	UI 29 2^{29}	UI 28 2^{28}	UI 27 2^{27}	UI 26 2^{26}	UI 25 2^{25}	UI 24 2^{24}

The system variables listed in Table 2.14.12 may take on 1.0 or 0.0 depending on whether the correspondign input signal is on or off.

Table 2.14.13 Variable Value

Input Signal	Variable Value
On	1.0
Off	0.0

(ii) All of the above 32 input signals (UI0-UI31) can be read collectively as a positive decimal value by specifying system variable #1032 on the right side of an expression.

$$\#1032 = \sum_{i=0}^{31} \# [1000+i] \times 2^{i}$$

(iii) It is impossible to assign a value to system variables #1000 to #1032 by specifying the value to the left side of an expression.

(b) Interface ouput signal

(i) To output an ON/OFF signal to any of the 32 output signals used for a macro program, specify the corresponding system variable #1100 to #1131 on the left side of an expression.

Table 2.14.14 Correspondence between Interface Output Signals and System Variables

System Variable	#1107	#1106	#1105	#1104	#1103	#1102	#1101	#1100
Output signal	UO 7 2^{7}	UO 6 2^{6}	UO 5 2^{5}	UO 4 2^{4}	UO 3 2^{3}	UO 2 2^{2}	UO 1 2^{1}	UO 0 2^{0}
System variable	#1115	#1141	#1113	#1112	#1110	#1111	#1109	#1108
Output signa	UO 15 2^{15}	UO 14 2^{14}	UO 13 2^{13}	UO 12 2^{12}	UO 11 2^{11}	UO 10 2^{11}	UO 9 2^{9}	UO 8 2^{8}
System variable	#1123	#1122	#1121	#1120	#1119	#1118	#1117	#1116
Output signal	UO 23 2^{23}	UO 22 2^{22}	UO 21 2^{20}	UO 20 2^{20}	UO 19 2^{19}	UO 18 2^{18}	UO 17 2^{17}	UO 16 2^{16}
System variable	#1131	#1130	#1129	#1128	#1127	#1126	#1125	#1124
Output signal	UO 31 2^{31}	UO 30 2^{30}	UO 29 2^{29}	UO 28 2^{28}	UO 27 2^{27}	UO 26 2^{26}	UO 25 2^{25}	UO 24 2^{24}

Assign 1.0 or 0.0 to the system variables listed in Table 2.14.14 to output on or off for the corresponding output signal.

Table 2.14.15 Variable Value

Output Signal	Variable Value
On	1.0
Off	0.0

If any value other than 1.0 or 0.0 is assigned to #1100 to #1131, the value is assumed to be as follows:

(Null) or a value less than 0.5	0.0
Other than the above	1.0

(ii) All of the above 32 output signals (UO0-UO31) can be output collectively with a positive decimal value by specifying system variable #1132 on the left side of an expression. The positive decimal value is converted to 32 bits for the output signals.

$$\boxed{\#1132 = \sum_{i=0}^{31} \#[1100+i] \times 2^{i}}$$

(iii) The on/off status of any output signal can be read (with 1.0, 0.0, or positive decimal value) by specifying system variable #1100 to #1132 on the right side of an expression.

(c) **Offset and work coordinate shift quantity**

(i) To read a tool offset, specify the corresponding system variable #12001 to #13499 on the right side of an expression.

(ii) To read a work coordinate system shift quantity, specify the corresponding system variable #14101 to #14112 on the right side of an expression.

(iii) To change any of the above system variables, assign a new value on the left side of an expression.

⟨Programming example⟩

(A) #116 = #12016

The contents of tool offset No. 16 are saved into common variable #116.

(B) #14101 = #4

The X-axis work coordinate system shift quantity is replaced by the contents of local variable #4.

(d) **Table 2.14.16** shows the correspondence between tool offset No. system variables and work coordinate shift quantity:

Table 2.14.16 Tool Offset No. System Variables (16 sets)

Offset Definition	1 program or shared by programs		Separate for 2 programs		Separate for 3 programs	
	Offset No.	System variable	Offset No.	System variable	Offset No.	Syatem variable
X-axis	01	#12001 (#2001)	01	#12001 (#2001)	01	#12001 (#2001)
	02	#12002 (#2002)	02	#12002 (#2002)	02	#12002 (#2001)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	16	#12016 (#2016)	16	#12016 (#2016)	16	#12016 (#2002)
Z-axis	01	#12301 (#2101)	01	#12301 (#2101)	01	#12301 (#2101)
	02	#12302 (#2102)	02	#12302 (#2102)	02	#12302 (#2102)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	16	#12316 (#2116)	16	#12316 (#2116)	16	#12316 (#2116)
Nose R Quantity	01	#12901 (#2201)	01	#12901 (#2201)	01	#12901 (#2201)
	02	#12902 (#2202)	02	#12902 (#2202)	02	#12902 (#2202)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	16	#12916 (#2216)	16	#12916 (#2216)	16	#12916 (#2216)
Control Point	01	#13201 (#2301)	01	#13201 (#2301)	01	#13201 (#2301)
	02	#13202 (#2302)	02	#13202 (#2302)	02	#13202 (#2302)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	16	#13216 (#2316)	16	#13216 (#2316)	16	#13216 (#2316)

NOTES

1. System variables #2001 to #2316 are for compatibility with the X series.
2. System variables #12001 to #13216 can be used on the i series only.
3. #2001 and #12001 are the same and either can be used. However, use the system variables for YASNAC i series if possible.
4. For multiple-program control, use parameter pm4020 D0 to specify whether to use the offsets separately for the programs.

   If pm4020 D0=0, offsets are shared by the sections.

   If pm4020 D0=1, different offsets are used in different programs.

Table 2.14.17 Tool Offset No. System Variables (99 sets)

Offset Definition	1 progarm or shared by programs		Separate for 2 programs		Separate for 3 programs	
	Offset No.	System variable	Offset No.	System variable	Offset No.	Syatem variable
X-axis	01	#12001 (#2001)	01	#12001 (#2001)	01	#12001 (#2001)
	02	#12002 (#2002)	01	#12002 (#2002)	02	#12002 (#2001)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	99	#12099 (#2099)	49	#12049 (#2049)	33	#12033 (#2033)
Z-axis	01	#12301 (#2101)	01	#12301 (#2101)	01	#12301 (#2101)
	02	#12302 (#2102)	02	#12302 (#2102)	02	#12302 (#2102)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	99	#12399 (#2199)	49	#12349 (#2149)	33	#12333 (#2133)
Nose R Quantity	01	#12901 (#2201)	01	#12901 (#2201)	01	#12901 (#2201)
	02	#12902 (#2202)	02	#12902 (#2202)	02	#12902 (#2202)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	99	#12999 (#2299)	49	#12949 (#2249)	33	#12933 (#2233)
Control Point	01	#13201 (#2301)	01	#13201 (#2301)	01	#13201 (#2301)
	02	#13202 (#2302)	02	#13202 (#2302)	02	#13202 (#2302)
	⋮	⋮ ⋮	⋮	⋮ ⋮	⋮	⋮ ⋮
	99	#13299 (#2399)	49	#13249 (#2349)	33	#13233 (#2333)

NOTES
1. System variables #2001 to #2399 are for compatibility with the X series.
2. System variables #12001 to #13299 can be used on the i series only.
3. #2001 and #12001 are the same and either can be used. However, use the system variables for YASNAC i series if possible.
4. For multiple-program control, use parameter pm4020 D0 to specify whether to use the offsets separately for the programs.

   If pm4020 D0=0, offsets are shared by the sections.

   If pm4020 D0=1, different offsets are used in different programs.

Table 2.14.18 Tool Offset No. System Variables (299 sets)

Offset Definition	1 program or shared by programs		Separate for 2 programs		Separate for 3 programs	
	Offset No.	System variable	Offset No.	System variable	Offset No.	System variable
X-axis	01	#12001 (#2001)	01	#12001 (#2001)	01	#12001 (#2001)
	02	#12002 (#2002)	02	#12002 (#2002)	02	#12002 (#2002)
	:	: :	:	: :	:	: :
	99	#12099 (#2099)	99	#12049 (#2099)	:	: :
	:	:	:	:	:	: :
	299	#12299	149	#12149	99	#12099 (#2099)
Z-axis	01	#12301 (#2101)	01	#12301 (#2101)	01	#12301 (#2101)
	02	#12302 (#2102)	02	#12302 (#2102)	02	#12302 (#2102)
	:	: :	:	: :	:	: :
	99	#12399 (#2199)	99	#12349 (#2199)	:	: :
	:	:	:	:	:	: :
	299	#12599	149	#12449	99	#12399 (#2199)
Nose R Quantity	01	#12901 (#2201)	01	#12901 (#2201)	01	#12901 (#2201)
	02	#12902 (#2202)	02	#12902 (#2202)	02	#12902 (#2202)
	:	: :	:	: :	:	: :
	99	#12999 (#2299)	99	#12949 (#2299)	:	: :
	:	:	:	:	:	: :
	299	#13999	149	#13049	99	#12999 (#2299)
Control Point	01	#13201 (#2301)	01	#13201 (#2301)	01	#13201 (#2301)
	02	#13202 (#2302)	02	#13202 (#2302)	02	#13202 (#2302)
	:	: :	:	: :	:	: :
	99	#13299 (#2399)	99	#13249 (#2399)	:	: :
	:	:	:	:	:	: :
	299	#13499	149	#13399	99	#13299 (#2339)

NOTES
1. System variables #2001 to #2399 are for compatibility with the X series.
2. System variables #12001 to #13499 can be used on the i series only.
3. #2001 and #12001 are the same and either can be used. However, use the system variables for YASNAC i series if possible.
4. For multiple-program control, use parameter pm4020 D0 to specify whether to use the offsets separately for the programs.

   If pm4020 D0=0, offsets are shared by the sections.

   If pm4020 D0=1, different offsets are used in different programs.

Table 2.14.19 Work Coordinate System Shift Quantity System Variable

		System Variable	
1st program	X	#14101	(#2000)
	Z	#14102	(#2100)
	C	#14103	:
2nd program	X	#14105	
	Z	#14106	
	C	#14107	
3rd program	X	#14109	
	Z	#14110	
	C	#14111	

NOTES
1. System variables #2000 to #2100 are for compatibility with the X series.
2. #2000 and #14101 are the same and either can be used. However, use the system variables for YASNAC i series if possible.

(d) Alarm message display

#3000 = ⟨Alarm No.⟩ (⟨Alarm message⟩) ;

⟨Alarm No.⟩: 4-digit alarm No. not being used by the controller (Use alarms 5000 to 5999.)

⟨Alarm message⟩: ASCII character string consisting of 32 or less (alphanumeric and special) characters

Specify the above instruction to issue an alarm. The alarm occurs after the preceding block has been executed.

(e) Clock

To read time, specify this system on the right side of an expression. To preset time, specify this system on the left side of an expression.

Table 2.14.20 Clock

Type	System Variable	Unit	Power-on	Count Condition	For multiple programs
Clock 1	#3001	1 ms	Preset to 0	Always	Common
Clock 2	#3002	1 s	Same as power-off	When STL signal is on	Separate
Clock 3	#3010	1 s	Same as power-off	When any STL signal is on among programs	Common

(f) Single-block stop and auxiliary function complete wait control

To nullify the single-block switch for the subsequent blocks, or to proceed to t he next block without waiting for the complete signal (FIN) of the auxiliary function (M,T) to be confirmed, assign the specific value to system variable #3003.

If ann auxiliary function is commanded in a state where check for the complete signal (FIN) of auxiliary function (M, T) is not waited for, distribution complete signal (DEN) is not output and FIN signal is not checked although M and T code output and M and T read output are made as usual.

The above state continues until a block, in which a value is assigned to #3003 so that check of auxiliary function complete signal will be waited for as usual, is executed. After such a block is executed, DEN is output and, at the same time, FIN is waited for.

Note that #3003 is cleared to 0 when reset.

Table 2.14.21 Single-block Stop and Auxiliary Function Complete Wait Control

#3003	Single-block Switch	Auxiliary Function Complete Signal (FIN)
0	Valid	Waited
1	Invalid	Waited
2	Valid	Not waited
3	Invalid	Not waited

Example of Programming

```
      :
    M10;        ← M10 code, MF output, DEN output, FIN waited
      :
    #3003=3
      :
    M20;        ← M20 code, MF output, FIN is not waited
      :
    M25         ← Ignored
      :
    #3003=0;    ← DEN output, FIN waited
      :
    M40;        ← M40 code, MF output, DEN output, FIN waited
      :
```

Single-block stop FIN is not waited for (from #3003=3 to #3003=0)

Indicated as "M" (M20 to #3003=0)

(g) Turning on and off feed hold; feedrate override; and exact stop check

To enable or disable feed hold, feedrate override, and exact stop check for the subsequent blocks, assign the following values to system variable #3004.

When reset, #3004 is cleared to 0.

Table 2.14.22 Feed Hold, Feedrate Override, and Exact Stop Check Control

#3004	Feed Hold	Feedrate Override	Exact Stop Check
0	Valid	Valid	Valid
1	Invalid	Valid	Valid
2	Valid	Invalid	Valid
3	Invalid	Invalid	Valid
4	Valid	Valid	Invalid
5	Invalid	Valid	Invalid
6	Valid	Invalid	Invalid
7	Invalid	Invalid	Invalid

(i) Feed hold

Feed hold is nullified from a block where #3004=1, 3, 5, or 7 is commanded until #3004=0, 2, 4, or 6 is commanded.

If the feed hold button on the operator panel is depressed while a block in which feed hold is nullified is being executed, feed hold is not accepted and the feed hold lamp will not light.

(ii) Feedrate override

Feedrate override is nullified from a block where #3004=2, 3, 6, or 7 is commanded until #3004=0, 1, 4, or 5 is commanded.

(iii) Exact stop check

Exact stop is not checked from a block where #3004=4, 5, 6, or 7 is commanded until #3004=0, 1, 2, or 3 is commanded.

Example of Programming

Special threading cycle (in incremental command)

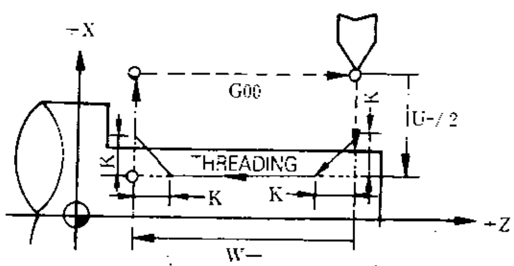
Macro call command

```
G65  P9093  U-···  W-···  K···  F······
```

- F → #9 : Lead of screw
- K → #6 : Unsigned radius
- W → #23 : Negative value
- U → #21 : Negative value, diameter

Macro Program

```
O9093 ;
M93 ;                                    ← 7-block buffering
#10=ROUND [#6] * 2 ;
#11=ROUND [#21] - #10 ;
#12=ROUND [#23] + #10 ;
#3003= 1 ;                               ← Single block nullified
G00  U#11 ;
#3004= 7 ;                               ← Feed hold, feedrate override, and completion of positioning: Invalid
G32  U-#10  W-#6  F#9 ;
G32  W#12 ;
G32  U#10  W-#6 ;
#3004= 0 ;
G00  U-#11 ;
G00  W-#23 ;
#3003= 0 ;
M92 ;
M99 ;
```



(h) RS-232C data output 1 (#3100)

To output messages or variable data to external equipment via the RS-232C data I/O interface, use system variable #3100.

(i) `#3100= ( ⟨message⟩ );`

Use this command to output the message enclosed in the control-out and control-in. CR and LF (carriage return and line feed) codes are automatically added to the end of the message.

⟨message⟩: ASCII character string consisting of 128 or less characters (alphanumeric and special characters)

`#3100= ( );`

Use this command to output only CR and LF (carriage return and line feed) codes.

(ii) `#3100 = [ ⟨variable⟩ ];`

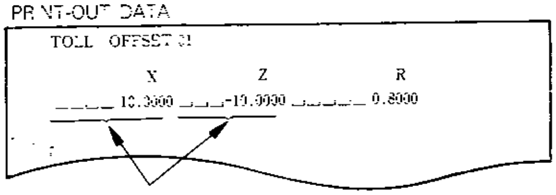
Use this command to output the value of «variable» as 9-digit decimal data. (Four digits before the decimal point and five after the decimal point).

⟨variable⟩: Local variable, common variable, or system variable

Example of Programming

```
#3100= (      ) ;······Return and line feed
#3100= (TOOL OFFSET 01) ;
#3100= (␣␣······X␣␣······Z␣␣······R) ;
#3100= [#2001] ;···=10.000mm    ┐
#3100= [#2101] ;···=−10.000mm   │ Values of arguments are
#3100= [#2201] ;···=0.800mm     ┘ written in comment.
#3100= (      ) ;
```

PRINT-OUT DATA



Up to 8-digit data including the sign can be output before the decimal point.

NOTES
1. The numbers will be rounded off at the fifth decimal place.
2. When there are more than six digits before the decimal point, ✻ (asterisk) is output.
3. If more than one RS-232 is output simutaneously, it is not output even in the multi-program control.
   If output is attempted from more than two-program control systems, an alarm (0239) will be output.
   Do not make simultaneous outputs by using the queuing function, etc.

(iii) Special codes that can be used in macro program

Special codes listed in **Table 2.14.23** can be used.

EIA code hole patterns marked with a star conform to Table 2.14.25 as standard.

Different hole patterns can be specified using the following parameters:

Hole pattern setting parameters

#4100	#
#4101	[
#4102	]
#4103	*
#4104	=
#4105	(
#4106	)
#4107	,

NOTE To specify a hole pattern to any of these parameters, find the binary code of the desired hole pattern, convert it to a decimal number, then set the decimal number to the parameter.

In the following example, hole pattern "152" is set:

8	7	6	5	4	∘	3	2	1
○			○	○	∘			

When set to 0, the above parameters conform to hole patterns in **Table 2.14.25**.

When any of the following codes are output from the controller for punching out, the upper code (UC) or the lower code (LC) is output immediately before it.

Codes output after UC	#, +, $, ?, ✻ =
Codes output after LC	@
Codes output after UC only when parameter is specified	(,)

Table 2.14.23 Special Codes

	Character or Symbol	Purpose	EIA Code									ISO Code								
			8	7	6	5	4	○	3	2	1	8	7	6	5	4	○	3	2	1
	SP	Comments				○		○				○		○			○			
★	(	Alarm message and comments				○	○	○		○				○		○	○			
★	)	Alarm message and comments		○			○	○		○		○		○		○	○			○
	+	Addition		○	○	○		○						○		○	○		○	○
	−	Subtraction		○				○						○		○	○	○		○
	:	Comments		○				○	○	○				○	○	○	○		○	
	/	Division			○	○		○			○	○		○		○	○	○	○	○
★	#	Variable	Specified by parameter									○		○			○		○	○
★	✻	Multiplication	○			○	○	○				○		○		○	○		○	
★	=	Equal sign	○				○	○	○			○		○	○	○	○	○		○
★	[	Left bracket	○		○	○		○				○	○		○	○	○		○	○
★	]	Right bracket	○		○			○		○		○	○		○	○	○	○		○
	$	Comments	○			○		○	○					○			○	○		
	@	Comments	○				○	○	○	○	○	○	○				○			
	?	Comments	○			○	○	○	○	○				○	○	○	○	○	○	○
	.	Decimal point		○	○		○	○		○	○			○		○	○	○	○	
★	,	Comma		○	○	○	○	○	○			○		○		○	○			

(i) Modal information

To read the modal commands that are instructed in previous blocks, use the following system variables on the right side of an expression.

These system variables cannot be described on the left side of an expression.

Table 2.14.24 Modal Commands and Macro System Variables

Modal Command	Macro System Variable
G code (group 01) to (group 31)	#4001 to #4031
E code	#4108
F code	#4109
Sequence number	#4114
Program number	#4115
S code (1)	#4119
T code	#4120

NOTES
1. M codes are non-modal information and cannot be read.
2. For both E (#4108) and F (#4109), value of the last F or E is saved. Therefore, #4108 and #4109 takes on the same value.

Example of Programming

Main program

```
G65 P9602 (Specification-of-argument) ;
```

Macro program

```
O9602 ;
#1 = #4001 ;  ← Saves G codes (G00 to G03) of group 01.
G00 X······Z······ ;
G01 Z······F······ ;
G03 X···Z···R··· ;
G00 Z··· ;
G#1 ;  ← Restores G codes of group 01.
M99 ;
```

(j) Position information

To read position information, use the following system variables.

These system variables cannot be described on the left side of an expression.

Table 2.14.25 Position Information

Position Information	Macro System Variable	Ready during Initiation
X-axis block end position (ABSIO) Z-axis block end position (ABSIO) C-axis block end position (ABSIO)	#5001 #5002 #5003	Possible
X-axis machine coordinate system positon (ABSMT) Z-axis machine coordinate system position (ABSMT) C-axis machine coordinate system position (ABSMT)	#5021 #5022 #50023	Possible (*)
X-axis current value display position (ABSOT) Z-axis current value display position (ABSOT) C-axis current value display position (ABSOT)	#5041 #5042 #5023	Possible (*)
X-axis skip signal position (ABSKP) Z-axis skip signal position (ABSKP) C-axis skip signal position (ABSKP)	#5061 #5062 #5063	Possible
X-axis offset Z-axis offset C-axis offset	#5081 #5082 #5085	Possible
X-axis servo position deviation Z-axis servo position deviation C-axis servo position deviation	#5101 #5102 #5105	Possible (*)

NOTE
1. For system variables marked with *, the position information is read after the preceding block has been completed.
2. If additional axis control is added, the axes and system variables may correspond differently from the table. Refer to the manual issued by the machine manufacturer.

Table 2.14.26

Abbr.	ABSIO	ABSMT	ABSOT	ABSKP
Meaning	Position of end point of preceding block	Command current position (same as machine coordinate display)	Command current position (same as current value display)	Position where skip signal goes on in G31 block
Coordinate system	Work coordinate system	Machine coordinate system	Work coordinate system	Work coordinate system
Tool position and nose R	Not included	—	Included	Included

NOTE The unit is metric or inch input unit specified at the time.

Table 2.14.27

	Unit	
	Micron	Submicron
Metric input	0.001mm	0.0001mm
Inch input	0.0001in	0.00001in
Degree input	0.001deg	0.0001deg

NOTES
1. If the skip signal is not turned on in the G31 block, the skip signal position indicates the end point of the G31 block.
2. If the skip signal is turned on in the G31 block, the G31 block end position indicates the skip signal position.

2.14.3 Specifying Variables

A variable is represented by a variable number or alphanumeric characters preceded by #.

(1) Direct specification by variable number

$\boxed{\#\,i}$ (i is the variable number)

(Example) #1, #101, #501, #2001

(2) Specification with an expression used for variable number

$\boxed{\#\,[\text{expression}]}$

(Example) # [#101], # [#501+1], # [#1/2]

2.14.4 Quoting Variables

A variable can substitute for a numeric value following an address.

⟨address⟩ #i or ⟨address⟩ − #i

With the above command, the value of the variable or its negative number (complement) can be used for the command value at the address.

(Example)

When G#30, #30=1.0	Equivalent to command "G01"
When X#101, #101=100.	Equivalent to command "G100."
When Z#103, #103=300.	Equivalent to command "Z300."
When F#140, #140=0.3	Equivalent to command "F0.3"

Therefore, G#30 X#101 Z#103 F#140 ; is equivalent to

G01 X100. Z300. F0.3 ;

(1) No variable can be quoted at address/, O, or N.

(Example)

/#5	"n" (n=1 to 9) in /n cannot be replaced with a variable.
O#100	O-number (program number) cannot be replaced with a variable.
N#200	N-number (sequence number) cannot be replaced with a variable.

(2) A variable number cannot be replaced with a variable.

(Example) To replace "10" in "#10" with #20, describe "# [#20]".
"##20" cannot be used.

(3) When a variable represents address data, any digits in the data less than the minimum setting unit for the address are rounded off.

(Example)

When X#1, #1=45.2346	X45.235mm (with 0.001-mm input unit)
When F #2, #2=0.2555	F0.256 mm/rev. (in F33 format)
When G04P #3, #3=5.37672	G04P5.377s
When M#4, #4= 2.7236	M03
When G#4, #4=2.7236	G03

(4) A number following an address can be replaced with an ⟨Expression⟩.

⟨address⟩ [⟨expression⟩] or ⟨address⟩ − [⟨expression⟩]

With the above command, the value of the ⟨expression⟩ or its negative number (complement) can be used for the command value at the address.

(5) For a constant that has no decimal point and described in brackets, a decimal point is assumed at the end of it.

2.14.5 Undefined Variables

Variables that have no value assigned yet are undefined variables. The variable value of an underfined variable is ⟨null⟩.

There are the following undefined variables:

(1) Local and common variables (#100 to #299) after power-on or reset

(2) Local variables to which no argument was assigned when a macro program was called.

(3) Local variables corresponding to the level of a macro program that has been returned by M99.

(4) Local and common variables to which no value was saved in a macro program.

(5) Common variables to which no value was written from the operation panel.

NOTE Variable #0 is used as the underfined variable that always has variable value ⟨null⟩. #0 cannot be described on the left side of an expression.

Meaning of ⟨null⟩

① If an underfined variable is quoted, the address is also disregarded.

When #2=⟨null⟩	G00X#2; is equivalent to G00;
When #2=0	G00X#2; is equivalent to G00X0

② If used in an expression, 0 is used for the value of an undefined variable, except when the value is replaced with ⟨null⟩.

When #2=⟨null⟩ and #3=#2;	#3=⟨null⟩
When #2=⟨null⟩ and #3=# [#2+#2];	#3=#0=⟨null⟩
When #2=⟨null⟩ and #3=#1*#2;	#3=0
When #2=⟨null⟩ and #3=#2+#2;	#3=0
When #2=⟨null⟩ and #3=#2/#2;	#3=0
When #2=⟨null⟩ and #3=5*#2;	#3=0
When #2=⟨null⟩ and #3=2−#2;	#3=2
When #2=⟨null⟩ and #3=5/#2;	Division error

③ If used in a conditional expression, 0 is used for the value of an underfined variable except for EQ and NE.

When #3=0 and #2=⟨null⟩	#3EQ #2 is not satisfied.
When #3=0 and #2=⟨null⟩	#3NE #2 is satisfied.
When #3=0 and #2=⟨null⟩	#3GE #2 is satisfied.
When #3=0 and #2=⟨null⟩	#3LT #2 is not satisfied.

2.14.6 Arithmetic Commands

Local variables, common variables, system variables, and constants can be combined with operators and functions in general arithmetic operations, and the results can be assigned to a specified variable.

Using a variable in an arithmetic operation means reading the required data from the internal variable data area. Substituting the result of an operation means writing the operation result data into the internal variable data area.

The write process is completed when one block has been executed.

The basic format of arithmetic operation is $\boxed{\#i = \langle expression \rangle}$.

The following operations and functions can be used:

(1) Definition and replacement of variables

#i=#j	Definition or replacement
#i=# [#j+#k]	Indirect specification

(2) Addition and subtraction

#i=#j+#k	Sum
#i=#j−#k	Differecence
#i=#jOR#k	Logical OR (Each bit of binary 32 bits is used as an operand.)
#i=#jXOR#k	Exclusive OR (Each bit of binary 32 bits is used as an operand.)

(4) Multiplication and division

#i=#j✻#k	Product
#i=#j/#k	Quotient
#i=#jAND#k	Logical AND (Each bit of binary 32 bits is used as an operand;)
#i=#jMOD#k	Remainder (#j and #k are rounded to an integer before the remainder is calculated. If #j is negative, #i is also regative.)

(4) Functions

#i=SIN [#j]	Sine (in degrees)
#i=COS [#j]	Cosine (in degrees)
#i=TAN [#j]	Tangent (in degrees)
#i=ATAN [#j] or #i=ATAN [#j/#k]	Reverse tangent
#i=SQRT [#j]	Square root
#i=ABS [#j]	Absolute value
#i=BIN [#j]	Conversion from BCD to binary
#i=BCD [#j]	Conversion from binary to BCD
#i=ROUND [#j]	Conversion into integer by rounding off
#i=FIX [#j]	Digits after the decimal point discarded
#i=FUP [#j]	Digits after the decimal point raised to a unit
#i=ASIN [#j]	Reverse sine
#i=ACOS [#j]	Reverse cosine
#i=LN [#j]	Natural logarithm
#i=EXP [#j]	Exponent based on e (=2.718...)

(5) Combination of operations

Operations and functions in (1) to (4) above can be combined. The operations take precedence in the following order: function, multiplication-division, and addition-subtraction.

(Example) #i=#j−#k✻SIN [#1]

①

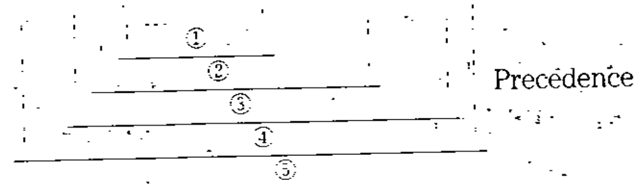
② Precedence

③

(6) Altering sequence of operations using brackets

Operations enclosed in brackets take precedence over other parts in the expression. The nesting level of brackets is up to five, including those for functions.

(Example) #i=SIN [[[#j+#k] ✱ #1+#m] ✱ #n]



NOTE The nesting level of brackets is up to five.

NOTES

1. **Constant in ⟨expression⟩ without decimal point**

   Constants having no decimal point described in ⟨expression⟩ are assumed to have the decimal point at the end of the constant. The range of constant values that can be used in ⟨expression⟩ is within ±99999999.99999999 (8 digits before the decimal point and 8 digits after.)

2. **ROUND Function**

   The ROUND function converts numbers into integers by rounding them to the nearest whole number. The digits are rounded off as follows:

   (a) If used in an operation command or conditional expression IF or WHILE, the digits after the decimal point are rounded off.

   (b) If used in address data, the digits less than the minimum setting unit of that address are rounded off.

   (Example 1) When #10=12.3758,
   #1=ROUND [#10]=12.0
   ROUND [#10] in IF [#10 GT ROUND [#10]] is 12.0.

   (Example 2) When #10=12.3758,
   G00 X [ROUND [#10]] (when minimum setting unit is 0.001 mm) is equivalent to G00 X 12.376.
   It is also equivalent to G00 X #10.

3. **Numerical data used in macro program**

   Numerical data in a macro program are processed in the floating-point system.

   $M * 2^E$ M: 1-bit sign + 52-bit binary data
   E: 1-bit sign + 10-bit binary data

2.14.7 Control Commands

The following two commands control the macro program flow:

- **Branch command**

```
IF [ ⟨conditional expression⟩ ] GOTO ⟨sequence No.⟩ ;
```

- **Repetition command**

```
WHILE [⟨conditional expression⟩] DO ⟨No.⟩ ;
    to
[ END ⟨No.⟩ ;
[ DO ⟨No.⟩;
    to
[ END ⟨No.⟩ ;
```

(1) Branch command

```
IF [ ⟨conditional expression⟩ ] GOTO ⟨sequence No.⟩ ;
```

The above statement orders that if «conditional expression» is satisfied, control should jump to the block of the specified sequence No. in the same program.

If ⟨conditional expression⟩ is not satisfied, control advances to the next block.

⟨sequence No.⟩: 5-digit integer, variable, or [⟨expression⟩]

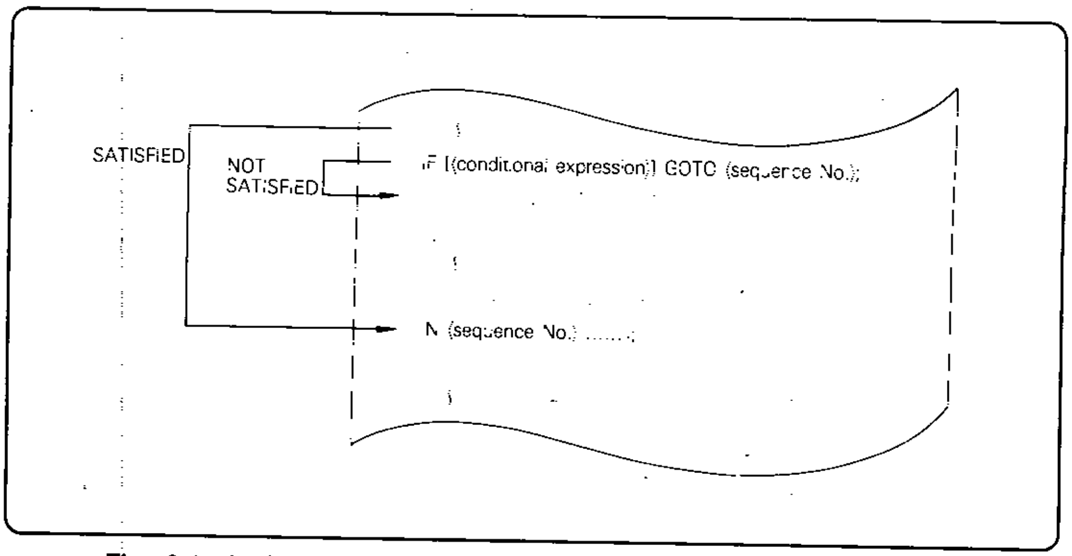


Fig. 2.14.6 Branch Command (Condition Satisfied and Not Satisfied)

If IF [(conditional expression)] is omitted, the command becomes a simple jump command.

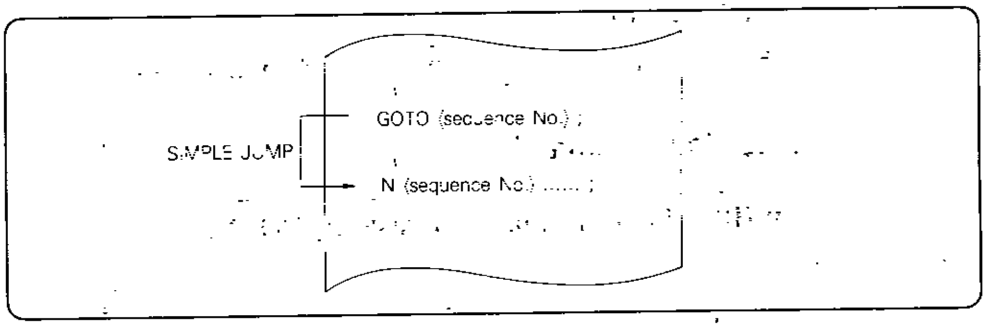


Fig. 2.14.7 Branch Command (Simple Jump Command)

NOTE As a rule, (sequence No.) specified for a branch command should be the first one in the block. Even if the second or later sequence is specified, execution of the jump destination block is started from the first line.
Branching towards the beginning of the program takes more execution time than branching towards the end.

In place of GOTO (sequence No.), a single block of NC statement or macro statement can be described. However, the following macro statements are excluded because of macro statement restictions:

Control command, RS-232C data output 2, status monitoring commands

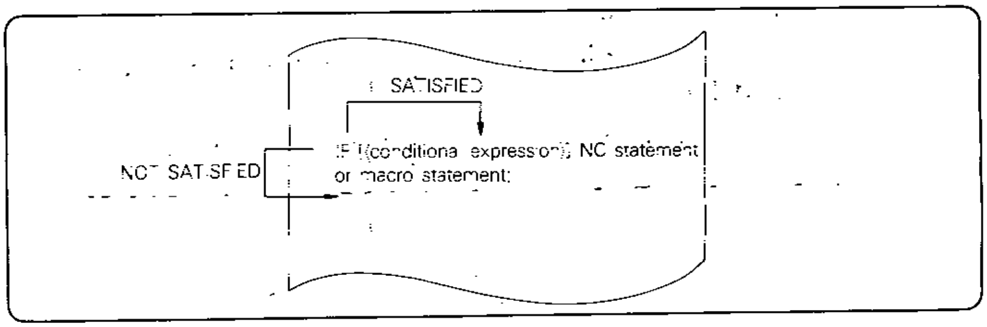


Fig. 2.14.8 NC Statement or Macro Statement Command

Table 2.14.28 lists the types of conditional expressions:

Table 2.14.28 Types of Conditional Expressions

Conditional Expression	Meaning
#iEQ#j	#i = #j
#iNE#j	#i ≠ #j
#iGT#j	#i > #j
#iLT#j	#i < #j
#iGE#j	#i ≧ #j
#iLE#j	#i ≦ #j

NOTE In place of #i and #j, constants or expressions can be used.

(2) **Repetition command**

WHILE [⟨conditional expression⟩] DO ⟨No.⟩ ;
to
[END ⟨No.⟩ ;

⟨No.⟩= 1, 2, 3

The above statement orders that operations of blocks between DO and END be repeated while ⟨conditional expression⟩ is satisfied. If ⟨conditional expression⟩ is not satisfied, control jumps to the block next to END.

When WHILE [⟨conditional expression⟩] is omitted, blocks between DO and END are infinitely repeated.

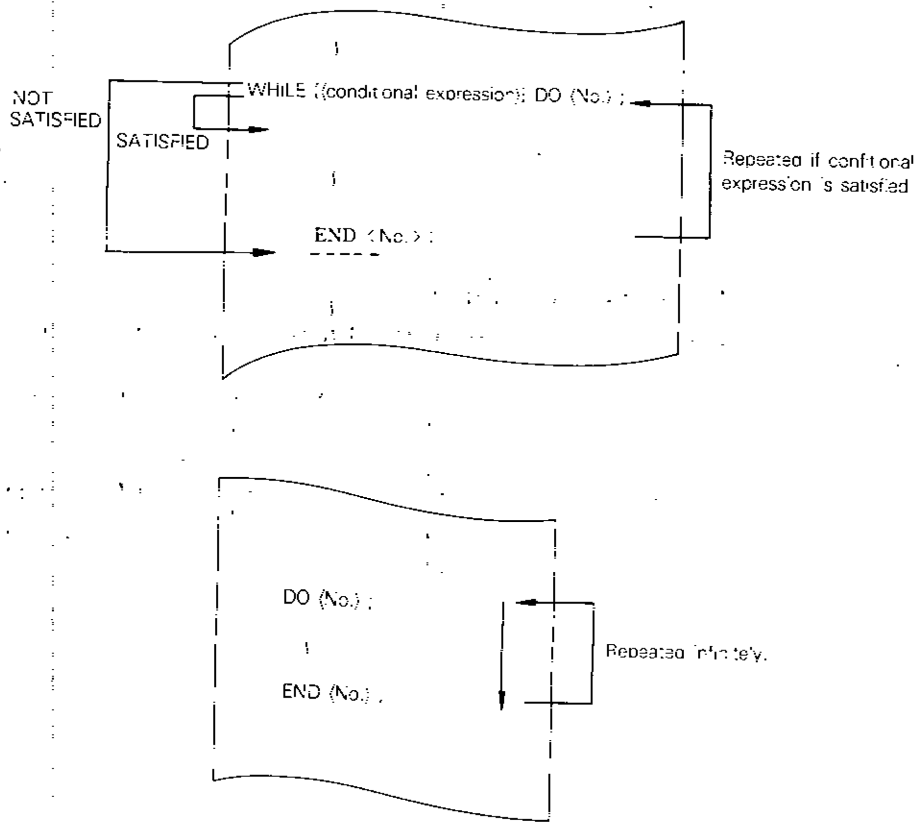


Fig. 2.14.9 Repetition Command

(a) DO must precede END.

DO1	END1
~	~
END1	DO1
○	×

○: Correct
×: Wrong

(b) (No.) in DO (No.) and END (No.) must be the same value, and must be specified as a pair on a 1-to-1 basis.

DO1	DO1	DO1	DO1
~	~	~	~
END1	END2	DO1	END1
⋮		~	~
DO2		END1	END1
~			
END2			
○	×	×	×

○: Correct
×: Wrong

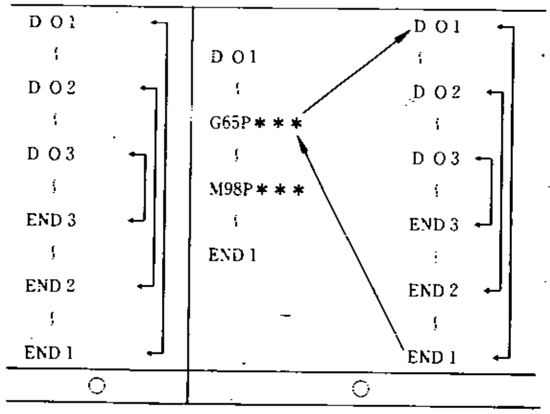
(c) The same (No.) can be used any number of times, provided that the blocks to be repeated do not overlap each other.

DO1	DO1	DO1	DO1
~	~	~	~
END1	DO1	DO1	DO2
⋮	~	~	~
DO1	END1	END1	END1
~	~	~	~
END1	END1	END1	END2
○	×	×	×

○: Correct
×: Wrong

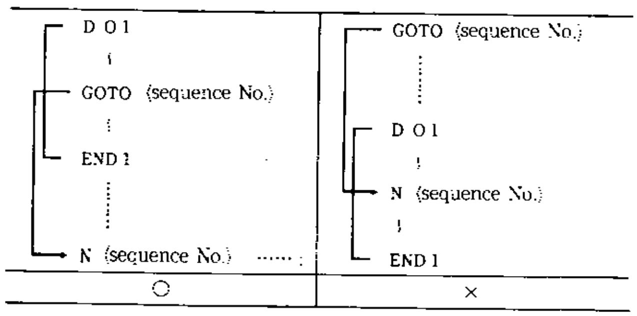
(d) The stacking in the DO-END loop is allowed up to a total of three in one macro program or subprogram.

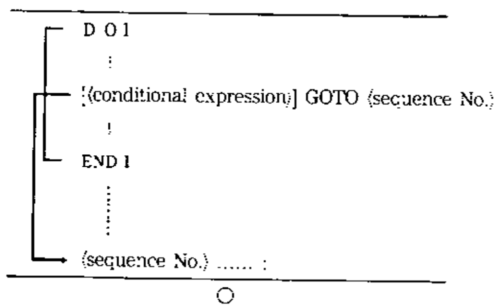
It is also possible to call a macro program or subprogram within the range from DO to END. The DO-END loop in the called program can also be used in up to triple-multiplexing.



○: Correct

(e) It is possible to jump off from inside the DO-END loop by using GOTO ⟨sequence No.⟩. On the contrary, it is impossible to jump into the DO-END loop by using GOTO ⟨sequence No.⟩.





○: Correct
×: Wrong

2.14.8 Entering macro programs

(1) Method of entering and editing macro programs

The method of entering and editing macro programs is just the same as the method used for normal NC part programs and subprograms.

There are no limits on the volume of a macro program. Total volume of NC part programs, subprograms, and macro programs to be stored must be within the NC part program memeory capacity.

(2) Macro program entry number

Mecro program entry numbers are classified by usage as shown in **Table 2.14.29.**

Table 2.14.29 Usage of Program

Program No.	Usage	Protect
O1 to O7999	Can be freely entered, erased, or edited	
O8000 to O8999	Edit protect and display protect can be independently applied by setting.	Protect 1
O9000 to O9999	Edit protect and display protect can be independently applied by parameters.	Protect 2
O10000 to O99999	Can be freely entered, erased, or edited	

Setting pm0020 D0, 0021 D0: Protect on/off in protect-1 area
Setting pm0022 D0, parameter pm3004 D0: Protect on/off in protect-2 area

2.14.9 RS-232C Data Output-2 (BPRNT, DPRNT)

In addition to RS-232C data output-1 described in Par. 2.14.2, the following macro commands can be executed:

Open command	(POPEN)
Data output command	BPRNT or DPRNT
Close command	(PCLOS)

These commands are used to output variables and characters through external equipment that has RS-232C interface.

(1) Open command (POPEN)

POPEN [a] ;

a: RS-232C number

This command outputs DC2 control code from the controller.

This command must be specified before a series of data output commands.

Specify 1 or 2 for the RS-232C number. If omitted, the 1st RS-232C interface is selected.

(Example)

POPEN ;	Open 1st RS-232C.
POPEN [2] ;	Open 2nd RS-232C.

(2) Data output command (BPRNT or DPRNT)

(a) BPRNT

BPRNT [a # b [c] .....] ;

c: Number of effective digits after decimal point

b: Variable

a: Character

This command executes the following function:

(i) Characters are output in the ISO code as they are specified. The following characters can be used:

- Alphabetic characters (A to Z)
- Numbers
- Special characters (✻, /, +, and −)

NOTE "✻" is output as the space code.

(ii) The value of a variable is handled as 2-word (32-bit) data, with the digits after the decimal point taken into account. The binary data value is output from the upper byte. Because all variables are stored in memory with the decimal point added, the number of effective digits after the decimal point must be specified enclosed in parentheses after the variable command.

(iii) After the commanded data has been output, the EOB code is output in the ISO code.

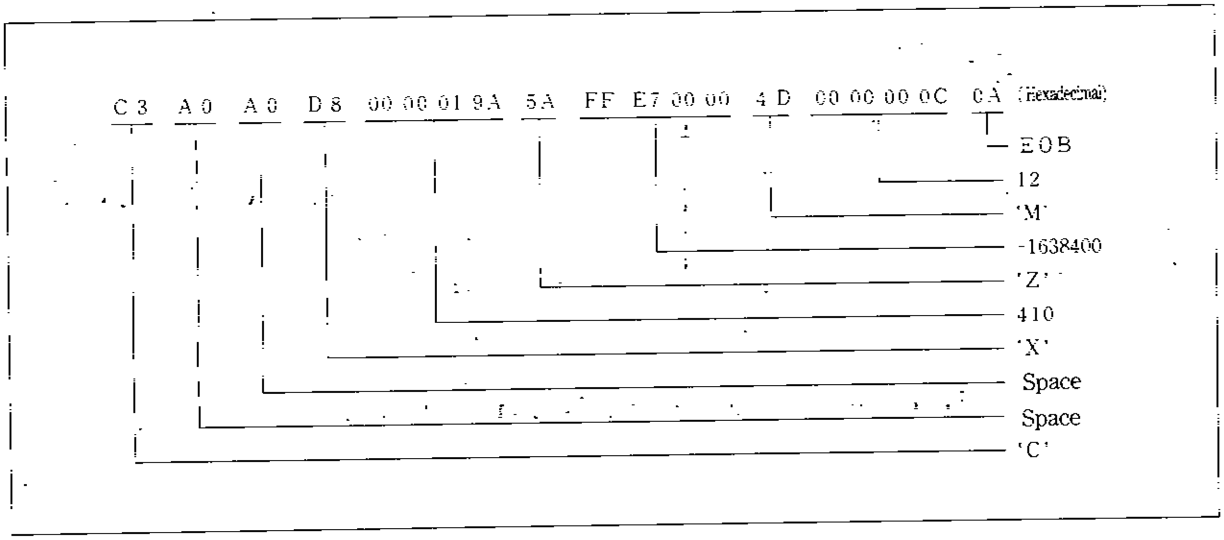
(iv) (Null) variables are assumed to be 0.

(Example)

BPRNT [C✻✻X#100 [3] Z#101 [3] Z#101 [3] M#10 [0]] ;

Value of variables: #100=0.40956
#101=−1638.4
#10=12.34

The above command outputs the following data:



(b) DPRNT

[a #b [c d] ......] ;

- a — Character
- b — Variable
- c — Number of digits before decimal point
- d — Number of digits after decimal point

This command executes the following function:

(i) Characters are output in the ISO code as they are specified, in the same way as for BPRNT command in (a).

(ii) The Value of a variable is output with a specified number of digits in ISO code from high-order digits. The decimal point is also output in the ISO code.

To output the value of a variable, specify the variable number after "#", then specify the numbers of digits before and after the decimal point enclosed in parentheses.

Parameter	pm4009 D2=0	Output space code.
	pm4009 D2=1	Output nothing.

If the specified number of digits after the decimal point is not 0, the specified number of digits is always output after the decimal point. If 0 is specified for the number of digits after the decimal point, the decimal point is not output.

Parameter	pm4009 D2=0	Output space code.
	pm4009 D2=1	Output nothing.

(iii) After the commanded data have been output, the EOB code is output in the ISO code.

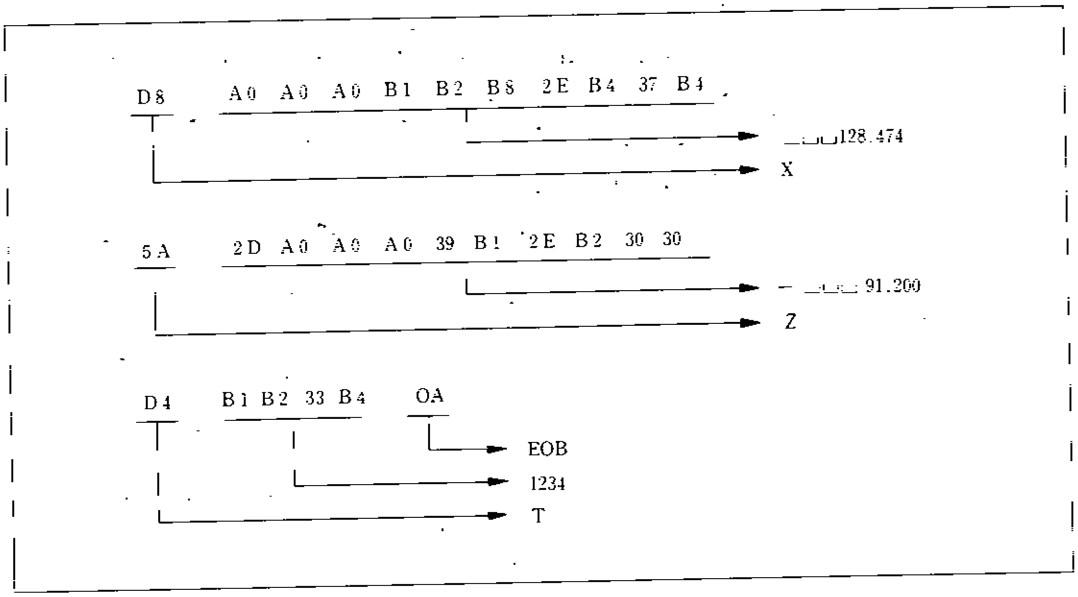
(iv) ⟨Null⟩ variables are assumed to be 0.

(Example)

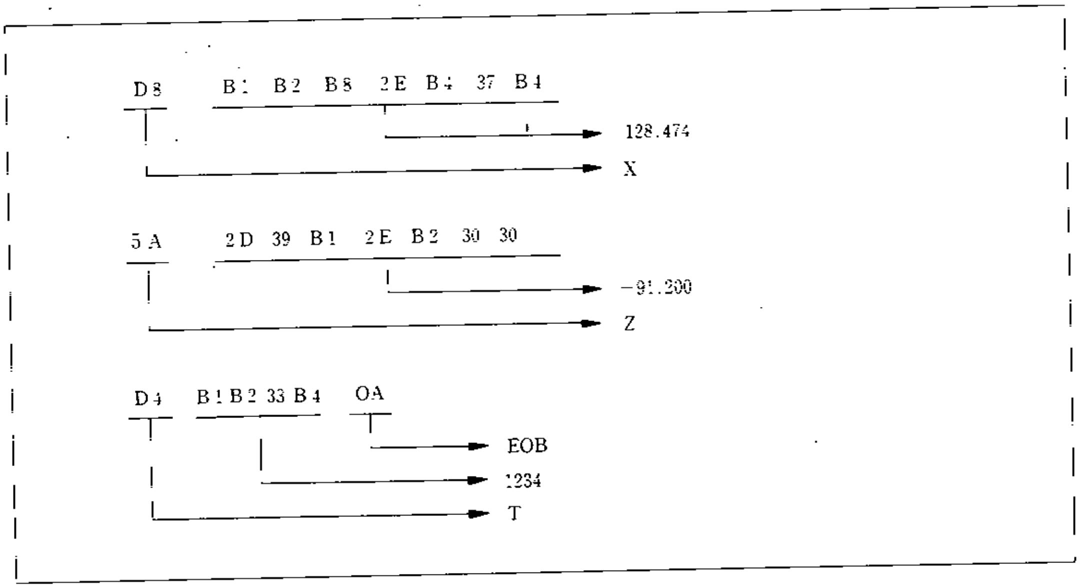
DPRNT [X#2 [53] Z#5 [53] T#30 [40]] ;

Value of variables: #2=128.47398
#5=−91.2
#30=1234.56

① When space code is to be output for + (Parameter pm4009 D2=0)



② When nothing is to be output for + (Parameter pm4009 D2=1)



(3) Close command (PCLOS)

Specify as follows to output the DC4 control code from the controller:

(Format) PCLOS [a] ;

a — RS-232C number

This command must be specified when all data output commands have been completed.

The RS-232C number is the same as that of POPEN.

(Example)

PCLOS;	Close 1st RS-232C.
PCLOS [2] ;	Close 2nd RS-232C.

NOTES

1. RS-232C data items

Set data (baud rate, etc.) for RS-232C interface by parameters.

2. Data output by BPRNT or DPRNT command

Set pm0004 D5 (use of parity bit for ISO tape output) to 0 (provided.) If the parameter is set to 1, output data are not guaranteed.

3. Data output by DPRNT command

To output data by the DPRNT command, specify whether to output space for leading zeros:

Parameter pm4009 D2 = 0	Output space for leading zeros in data output by DPRNT command.
Parameter pm4009 D2 = 1	Output nothing for leading zeros.

4. Specifying open command (POPEN) and close command (PCLOS)

Open command (POPEN) and close command (PCLOS) need not be specified in succession.

Once the open command is issued, it need not be specified again until the close command is used.

5. Notes on data output

If the controller is reset during data output, the output command is stopped and the subsequent data are lost.

If the controller is to be reset by M30 at the end of the program that outputs data, specify the close command at the end of the program so that M30 will be executed after all data have been output.

6. Open and close commands must be specified as a pair

The open and close commands must be specified as a pair. Do not specify the close command alone when there is no open command.

7. RS-232C during multi-program control

More than one RS-232C interface cannot be used at the same time even in a multi- program system. If attempted, alarm 0239 occurs. Use the wait function to avoid simultaneous use of multiple interfaces.

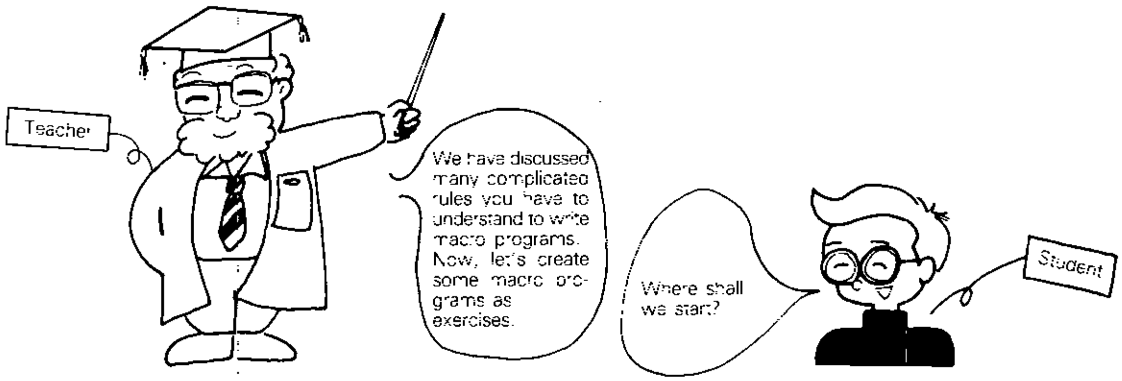
2.14.10 Macro Program Alarm Numbers

Table 2.14.30 lists the alarm numbers and the causes related to macro programs:

Table 2.14.30 Macro Program Alarm Numbers

Alarm No.	Explanation	Alarm No.	Explanation
#0210	Macro constant overflow: Constant in macro program exceeded range.	#0221	Macro division by 0: Division by 0 was attempted in macro program.
#0211	Macro command cancel error: There are too many G67 cancel codes.	#0222	Square root of negative number: Square root of negative number was required.
#0212	Macro format error: There is error in format.	#0223	Floating-point data overflow: Floating-point data exceeded range.
#0213	Undefined variable No. used: Specified value is not defined as variable number.	#0224	G66-M99 command error: Axial movement at return by M99 is commanded in modal call (G66.)
#0214	Variable unusable on left side: Variable in assignment statement is unusable for assignment.	#0225	Macro system error: Operation stack overflowed.
#0215	Bracket stacking exceeded. Too many brackets are nested.	#0226	ASIN, ACOS, LN, SQRT error: Operation result by ASIN, ACOS, LN, or SQRT function exceeded range.
#0216	Macro call stacking exceeded. Too many cacro calls are nested.	#0227	Integer conversion overflow: Integer conversion result exceeded range.
#0217	DO-END format error: DOs and ENDs do not correspond on 1-to-1 basis.	#0228	BCD input data overflow: Input data for BCD function exceeded range.
#0218	Bracket command format error: Number of left and right brackets mismatched.	#0229	BIN format error: There is in BIN function form at.
#0219	DO-END NO. range error: In DOm command, m is out of range from 1 to 3.	#0230	EXP output data overflow: Operation result by EXP function exceeded range.
#0220	GOTO NO. format error: In GOTOn command, n is out of range, or n was not found.	#0239	RS-232C multi-program command error: RS-232C is used by another section.

2.14.11 Macro Program Examples



: The G92 straight threading cycle would be easier for the first step.

The following is an example of the incremental G92 command:

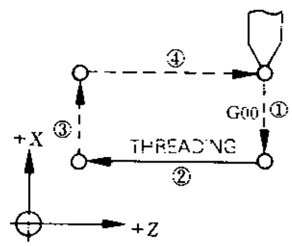
(P1)

```
G92 U-50. W-60. F6.0 ;
```

The controller divides the command as follows to execute it. Thread cut is not performed in this example.

(P2)

```
① G00 U-50. ;
② G32 W-60. F6.0 ;
③ G00 U-50. ;
③ G00 W-60. ;
```



First, these lengths of travel and the lead of the screw may all be converted into variables.

: You mean #1 to #33 local variables?
But there are type I and type II local variables.

: When variables are few, type I is better because addresses U, W, and F can be used so that argument specification is easy to understand.

: We will use it.
When type I is used, the table of variables becomes like this, doesn't it?

(P3)

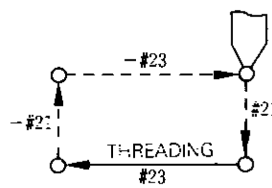
U-50.	W-60.	F-6.0
↓	↓	↓
#21	#23	#9

: Right.
Now, rewrite the previous program (P2) using these variables.

: Yes, I will.

(P4)

```
① G00 U#21 ;
② G32 W#23 F#9 ;
③ G00 U-#21 ;
④ G00 W-#23 ;
```



Is this all right?

: Correct. Add

M99 ;

after that. Now, a macro program is completed.

: Completed? It's very simple.

: Now, connect this macro call and the macro program in a complete form.

: Yes. Uhhhm.... With macro program number O9093, the macro call command is:

(P5)

```
G65 P9093 U-50.
W-60. F6.0 ;
```

The macro program is:

(P6)

```
O9093 ;
G00 U#21 ;
G32 W#23 F#9 ;
G00 U-#21 ;
G00 W-#23 ;
M99 ;
```

I arbitrarily assigned program number O9093 to the macro grogram.

: That looks OK.

****


: Sir, that is strange. With this program, addresses W and F must be specified every time.


: That's right. Good point.
In an ordinary canned cycle, once point-W and point-F are specified, those values are retained and only U needs to be specified for the second and subsequent operations.


: Isn't there any good idea?


: Yes, there is. Common variables (#100 to 559) are useful in this situation. You can create a macro program where the values of W and F are specified by common variables in the first place.


: I see. Now, I divide the macro program into two parts as follows.

```
O9000 ;
#100=#23 ;
#101=# 9 ;
M99 ;
```

```
O9093 ;
G00  U#21 ;
G32  W#100  F#101 ;
G00  U-#21 ;
G00  W-#100 ;
M99 ;
```

```
G65   P 9000  W-60. F6.0 ;
G65   P 9093  U-50. ;
G65   P 9093  U-51.4 ;
G65   P 9093  U-52.6 ;
G65   P 9093  U······ ;
  ⋮
```

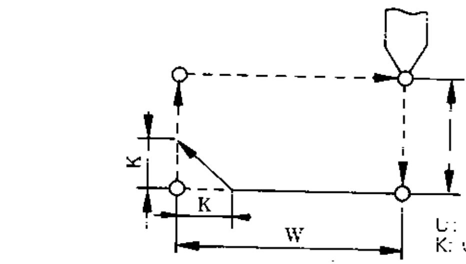

: Very good.

****

: Sir, I wish to program thread cut.

: Good.
Why won't you specify the thread cut width by address K?

: Oh, yes.



Then the macro call command is:

```
G65  P9000  W-60.  K4.8  F6.0 ;
G65  P9093  U-50. ;
G65  P9093  U-51.4 ;
G65  P9093  U…… ;
  :
```

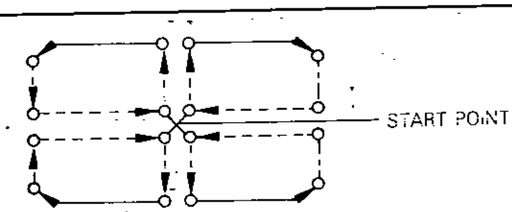
And the macro program is:

```
O9000 ;
#100=#23 ;
#101=# 9 ;
#102=ABS [# 6] ;
M99 ;
```

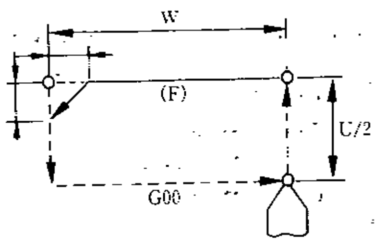
```
O9093 ;
#10=ROUND [#102] * 2 ;
#11=ROUND [#21] +#10 ;
#12=ROUND [#100] +ROUND [#102] ;
G00  U#21 ;
G32  W#12  F#101 ;
G32  U#10  W-#102 ; ←Thread cut
G00  U-#11 ;
G00  W-#100 ;
M99 ;
```

Is this all right?

: Well, it's very good in theory.
But in practice, it is advisable to add #3000 for single-block invalidation control and #3004 for feed hold invalidation control to prevent operation fault. And one more thing. This cycle can perform threading only in the U- or W-direction. T could have been made capable of threading in any of the four directions in the figure.



: Let me see.....
How about this one?



U and W can be signed.
K must be unsigned.
U = #21 (diameter)
W = #23
K = #6 (radius)
F = #9

The macro call command is:

```
G65 P9000 W-45. K4.0 F5.0 ;
G65 P9093 U40. ;
G65 P9093 U41.4 ;
G65 P9093 U...... ;
   :
```

The macro program is:

```
O9000 ;
#100=#23 ;  ←——— W
#101=#9 ;  ←——— F
#102=ABS [#6] ;  ←——— |K|
M99 ;
```

```
O9093 ;
#3003= 1 ;  <---------- Single-block nullified
M93 ;  <---------- 7-block buffering
#10=ROUND [#102] * 2 ;
IF [ABS [#21] LT#10] GOTO4 ;
IF [#21  GT   0] GOTO1 ;
IF [#21  EQ   0] GOTO4 ;
#11=ROUND [#21] +#10 ;  <------ U is negative
#12=#10 ;
GOTO2 ;
N1   #11=ROUND [#21] -#10 ;  <---- U is positive
#12=-#10 ;
N2   #13=ROUND [#102] ;
IF [ABS [#100] LT#13] GOTO4 ;
IF [#100GT0] GOTO3 ;
IF [#100EQ0] GOTO4 ;
#14=ROUND [#100] +#13 ;  <------ W is negative
#15=-#13 ;
GOTO5 ;
N3   #14=ROUND [#100] -#13 ;  <---- W is positive
#15=#13
GOTO5 ;
N4   #3000=499 (MACRO INPUT ERR.) ;  <---------- Fault display
N5   G00  U#21 ;
#3004= 7 ;  <---------- Feed hold, feedrate override, and completion of positioning nullified.
G32  W#14  F#101 ;
G32  U#12  W#15 ; — Screw cut
#3004= 0 ;  ----------
G00  U-#11 ;
G00  W-#100 ;
M92 ;
#3003= 0 ;
M99 ;
```

: Ah, you've made it such that a fault will occur if:

$$\begin{cases} \text{U or W is 0, and} \\ |U/2| < K \text{ or } |W| < K \end{cases}$$

very good.

: Thank you, Sir.

2.15 AUTO MEASUREMENT FUNCTION

2.15.1 Skip Function (G31) *

G31 X(U).......Z(W).......(F(E).......) ;

The above command executes special linear interpolation. If the skip signal input is turned on during move in this interpolation, the rest of the move commands in the block are skipped and the next block is started.

(1) Move after skip signal is turned on

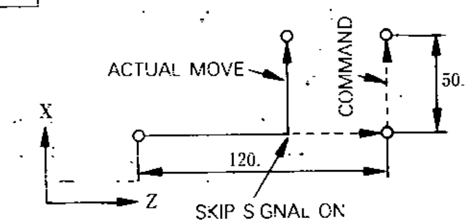
Move after skip signal depends on the next block command.

(a) If the next block command is incremental

The move starts from the position where the preceding move is aborted by the skip signal, according to the incremental command in the next block is started.

Example of Programming

```
G31  W120. ;
G01  U100. ;
```

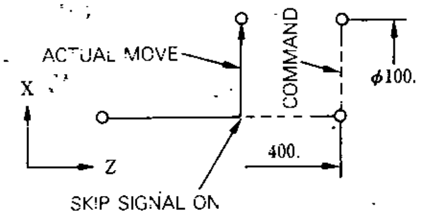


(b) If the next block command is absolute and only one axis is specified

The specified axis moves to the position of the commanded coordinate. The other axis is held at the position where the preceding move was aborted by the skip signal.

Example of Programming

```
G31  Z400. ;
G01  X100. ;
```

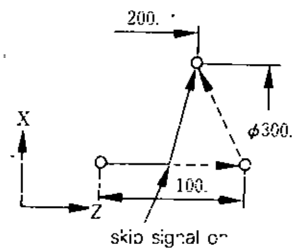


(c) If the next block command is absolute and both axes are specified

The axes move to the commanded position from the position where the skip signal was turned on.

Example of Programming

```
G31  W100. ;
G01  X300.   Z200. ;
```



(2) G31 is non-modal G code

If no skip signal is input after G31 block is executed, operation is stopped at the end of the block and alarm 0491 is displayed.

For the 2nd and 3rd sections, alarms 0492 and 0493 are displayed.

(3) G31 block feedrate

The feedrate in G31 block can be determined by either of the following methods. Select one by setting the parameter (pm2001 D0.)

(a) Specify with F in the same way as in ordinary programs.

(b) Specify predetermined feedrate to parameter pm2440.

(4) If skip signal goes on

The coordinates where the skip signal goes on are saved in memory automatically as parameter data.

pm0811: X coordinate
pm0812: Z coordinate

These data can be used as system variables in the macro program.

NOTES ON G31

1.Action to be taken if skip signal is not turned on

It is possible to set automatic advance to the next block in case the skip signal is not turned on after G31 is executed. To do this, set Setting pm0007 D2 to 1.

2.Specify G40 before G31

Specify G40 for nose R offset cancel before G31. Otherwise, alarm pm0182 occurs.

3.If G31 is issued while skip signal is on

If G31 is issued while the skip signal is on, the G31 block is not executed at all and the next block is started immediately.

2.16 STORED STROKE LIMIT CHECK

2.16.1 Stored Stroke Limit A

This function checks to see that the current values of the axes are not in the prohibit area during manual or auto operation. The prohibit area is specified by a parameter or the G36 or G38 command. If any axis is in the prohibit area, operation is stopped and alarm "2011" to "2018" is displayed.

By setting pm5000 D6 = 1, only stroke check monitor output (#3640, #3641) can be output without causing the alarm.

(1) Prohibit area

Outside of the area specified by the 1st prohibit area (stored stroke limit A) parameter is prohibited. In general, this check function can be used in place of over-travel check. Specify upper limit A_1 and lower limit B_1 by parameters.

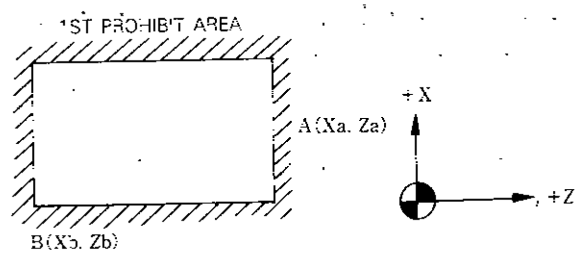


Fig. 2.16.1 Stored Stroke Limit A

(2) Setting stored stroke limit check for 1st prohibit area

Whether to check stroke A (whether to perform 1st prohibit area stored stroke limit check) can be specified on each axis.

Table 2.16.1 1st Prohibit Area Stored Stroke Limit Check On/Off

Axis	1st axis	2nd axis	3rd axis	4th axis	5th axis	6th axis	7th axis	8th axis
Parameter	pm6002 D0	pm6002 D1	pm6002 D2	pm6002 D3	pm6002 D4	pm6002 D5	pm6002 D6	pm6002 D7
Explanation of bit	0: Perform limit check.							
	1: Do not perform limit check.							

(3) Parameters and setting for specifying area

Table 2.16.2 Parameters and Setting for Specifying 1st Prohibit Area

Axis name	1st axis	2nd axis	3rd axis	4th axis	5th axis	6th axis	7th axis	8th axis
Boundary value (+): Point A	pm6901	pm6902	pm6003	pm6004	pm6005	pm6006	pm6007	pm6008
Boundary value (−): Point B	pm6911	pm6912	pm6913	pm6914	pm6915	pm6916	pm6917	pm6918

2.16.2 Stored Stroke Limit B

(1) Specifying area

An area can be specified by setting, or by G36 or G38 in the program. Whether the inside or outside of the area is to be prohibited is specified by a parameter.

The 2nd to 5th prohibit areas are stored storke limit B.

(a) `G36 U......W...... I......K...... (P......) ;` (Setting 2nd prohibit area)

- U......W...... : Coordinate of point C
- I......K...... : Coordinate of point D
- (P......) : 3rd to 5th prohibit areas can be set by specifying 3 to 5 for P.

The above command sets the area, and at the same time, turns on this area check.

`G37 (P......) ;`

- (P......) : If P is specified, turn off the prohibit area specified by P.

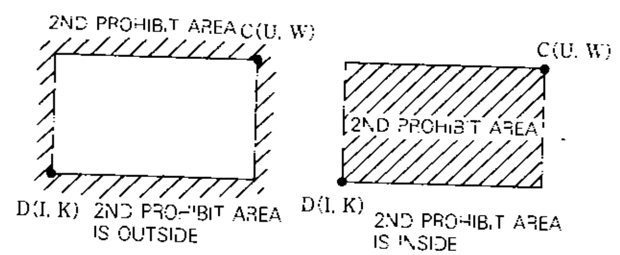


Fig. 2.16.2 Stored Stroke Limit B

(b) `G38 U......W...... I......K...... ;` (Setting 3rd prohibit area)

- U......W...... : Coordinate of point E
- I......K...... : Coordinate of point F

The above command sets the area, and at the same time, turns on this area check.

This area check is turned off by a single block of G39;

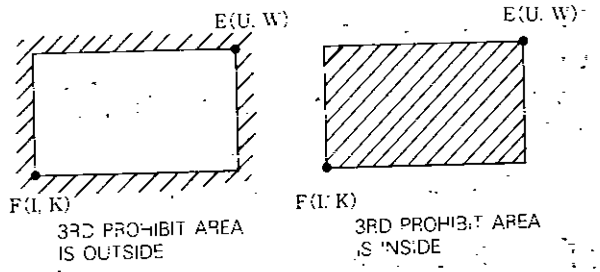


Fig. 2.16.3 Stored Stroke Limit C

(2) Setting number for specifying area

Table 2.16.3 Setting Number for Specifying Area

Prohibit area boundary value	1st axis	2nd axis	3rd axis
2nd prohibit (+): Point C	pm0831	pm0832	pm0833
2nd prohibit (−): Point D	pm0834	pm0835	pm0836
3rd prohibit (+): Point E	pm0837	pm0838	pm0839
3rd prohibit (−): Point F	pm0840	pm0841	pm0842
4th prohibit (+)	pm0843	pm0844	pm0845
4th prohibit (−)	pm0846	pm0847	pm0848
5th prohibit (+)	pm0849	pm0850	pm0851
5th prohibit (−)	pm0852	pm0853	pm0854

NOTE Points C and E specify a boundary value (+)
Points D and F specify a boundary value (−)

(3) Specifying check axis

Specify the axis to be checked for stored stroke limit in the 2nd to 5th prohibit area by the parameter. (Up to 3 axes can be checked.)

Table 2.16.4 Parameters for Specifying Check Axes in 2nd to 5th Prohibit Area

Check Area	Stroke Limit Check Axis No.		
	No.1	No.2	No.3
2nd prohibit area	pm6111	pm6112	pm6113
3rd prohibit area	pm6114	pm6115	pm6116
4th prohibit area	pm6117	pm6118	pm6119
5th prohibit area	pm6120	pm6121	pm6122

(Example) 1 = 1st X-axis
2 = 1st Z-axis
3 = 2nd X-axis
4 = 2nd Z-axis

(4) Setting number for specifying inside-outside for 2nd to 5th prohibit areas

Table 2.16.5 Inside Outside Specification for 2nd to 5th Prohibit Areas

Check Area	Setting Parameter No.	Explanation
2nd prohibit area	pm0008 D4	0: Inside is prohibited. 1: Outside is prohibited.
3rd prohibit area	pm0008 D5	
4th prohibit area	pm0008 D6	
5th prohibit area	pm0008 D7	

(5) Setting 2nd prohibit area check

Check in the 2nd to 5th prohibit areas can be turned on and off by setting.

Table 2.16.6 On/off of 2nd Prohibit Area

Check Area	Setting Parameter No.	Explanation
2nd prohibit area	pm0008 D0	0: Off 1: On
3rd prohibit area	pm0008 D1	
4th prohibit area	pm0008 D2	
5th prohibit area	pm0008 D3	

The setting data are automatically rewritten when G36-G39 is specified. Therefore, the check is turned on or off according to the last G code command or setting operation.

The 1st prohibit area check is always on.

(6) Area setting coordinate system

All the coordinates for prohibit area check muat be set as absolute coordinates.in the machine coordinate system. Specify the distance from the (1st) reference point. ["1"=minimum output (move) unit] Therefore, the check function cannot be used before manual or auto return to reference point is executed after power-on.

(7) Starting area check

Area check is started immediately after the first manual or auto return to the reference point is completed after power-on.

If the reference point is in the ptohibit area, immediately a stored stroke limit fault occurs. If this occurs, turn off the area check and rewrite data.

(8) Stored stroke limit fault

If the tool enters a prohibit area, it stops near the boundary and stored stored stroke limit error occurs. The tool can be taken back manually.

(9) Stroke limit B in multi-section system

(a) Concept of stroke limit B in multi-section system

Even in a multi-section system, four (2nd to 5th) prohibit areas are held for stroke limit B.

Which axes compose which area is determined by parameters pm6111 to pm6121 (stroke limit check axis number.)

For example, Z1, Z1, X2, and Z2 axes are used in 2-section system, the parameters can be set as follows: pm6111=1 (X1,) pm6112=2(Z1,) pm6114=3(X2,) and pm6115=4 (Z2.) Then the structure is that the 2nd prohibit area is X1-Z1 space and the 3rd prohibit area is X2-Z2 space. This means that the 3rd prohibit area is the 2nd prohibit area of the 2nd section. (**See Fig.2.16.4.**)

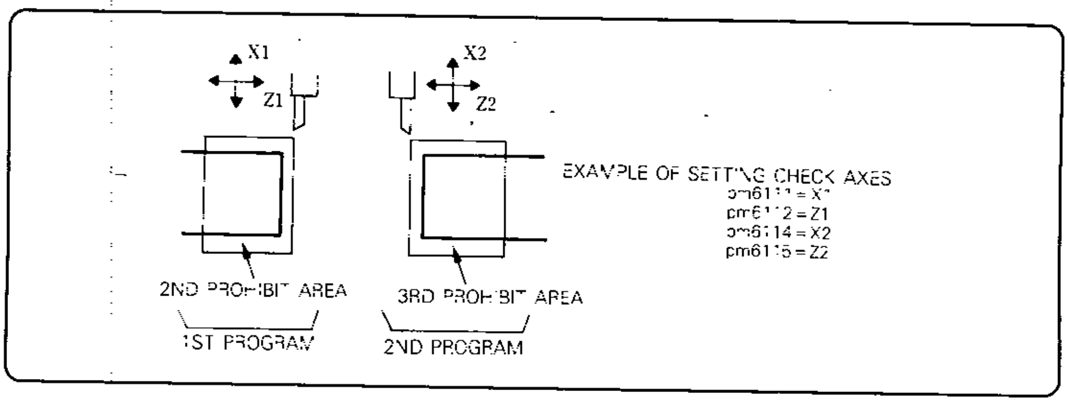


Fig. 2.16.4 Example of Setting Stored Stroke Limits B and C in Multi-section System

On/off of the above check areas can be determined by setting. The 2nd to 5th areas can be assigned to any section by setting.

(b) On/off command in program

To specify the area boundary and on/off of the check in a part program, specify as follows:

(i) With G36P command

The 2nd to 5th check areas can be directly set with P numbers by a command from any section. To determine an area and turn on check, describe as follows:

| G36 U...W...I...K...P... ; |

To turn off check, describe as follows:

| G37 P... ; |

(Example)

| G36......P3 ; | in 1st area program ← 3rd area on

| G36......P2 ; | in 2nd area program ← 2nd area on

(ii) Without P command in G36 or G38 command

The 2nd to 5th areas are determined by the number of programs and the program where the G36 or G38 command is executed:

- **In 2-program system**

Program		Prohibit area
1st program	G36... ; G37 ;	2nd prohibit area
	G38... ; G39 ;	4th prohibit area
2nd program	G36... ; G37 ;	3rd prohibit area
	G38... ; G39 ;	5th prohibit area

- **3-program system**

Program		Prohibit area
1st program	G36... ; G37 ;	2nd prohibit area
	G38... ; G39 ;	5th prohibit area
2nd program	G36... ; G37 ;	3rd prohibit area
	G38... ; G39 ;	5th prohibit area
3rd program	G36... ; G37 ;	4th prohibit area
	G38... ; G39 ;	5th prohibit area

The one specified last takes effect.

NOTES

1. Inconsistency between the axes specified by parameters pm6111 to pm6122 and machine configuration is not checked. Take care when setting.
2. In "G36 U...W...I...K... (P...) ;", "U..." and "I..." change position on X axes (X1, X2, ...) among axis numbers specified by parameters pm6111 to pm6121. "W..." and "K..." change Z axes (Z1. Z2, ...) among them.

   Therefore, the name of a prohibit area must be begin with X or Z.

NOTES

1. Prohibit area includes the boundary.
2. Prohibit areas can overlap each other. (**Fig. 2.16.5.**)

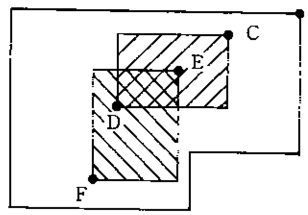
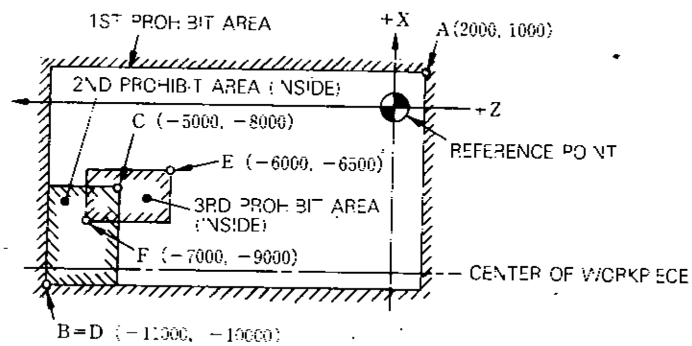


Fig. 2.16.5

3. Area check is executed at machine coordinates if machine lock is on.

Example of setting area



	Parameter/setting	Contents	
Inside/Outside	pm0008D4	0	(2nd)
	pm0008D5	0	(3rd)
2nd Area	pm0831	−5000	C
	pm0832	−8000	
	pm0834	−11000	D
	pm0835	−10000	
3rd Area	pm0837	−6000	E
	pm0838	−6500	
	pm0840	−7000	F
	pm0841	−9000	
1st Area	pm6901	2000	A
	pm6902	1000	
	pm6911	−11000	B
	pm6912	−10000	

2.17 MULTI-SECTION SUPPORT FUNCTION (FOR MULTI-SECTION OPTION ONLY)

2.17.1 Section Synchronization Command *

Use this function to wait completion of operations of other programs in a multi-program system to synchronize program execution.

(1) **Pause command**

Table 2.17.1 lists M and G codes that can be used for synchronization. Specify whether to use M code or G code by parameter pm4027 D0.

Table 2.17.1 Pause Commands

Pause Command		Explanation
M code (pm4027 D0 = 0)	G code (pm4027 D0 = 1)	
M950 to M999	G980 to G999	Synchronize 1st and 2nd programs.
M900 to M949	G960 to G979	Synchronize 1st and 3rd programs.
M850 to M899	G940 to G959	Synchronize 2nd and 3rd programs.
M800 to M849	G920 to G939	Synchronize three programs.

(2) **Execution of pause commands**

Two programs are synchronized as follows. The M code or G code on Table 2.17.1 is specified in the programs to be synchronized. The program where the command is executed first is stopped at the command. Operation of this program is paused until the same M or G code command is executed in the other program. When the same command is executed in the waiting program, the pause state of the suspended program is cleared and program is restarted.

Three programs are synchronized as follows. The command stops the first two programs. Operations of these programs are suspended until the same M or G code command is executed in the third program. When the same command is executed in the last program, operations of the three programs are restarted.

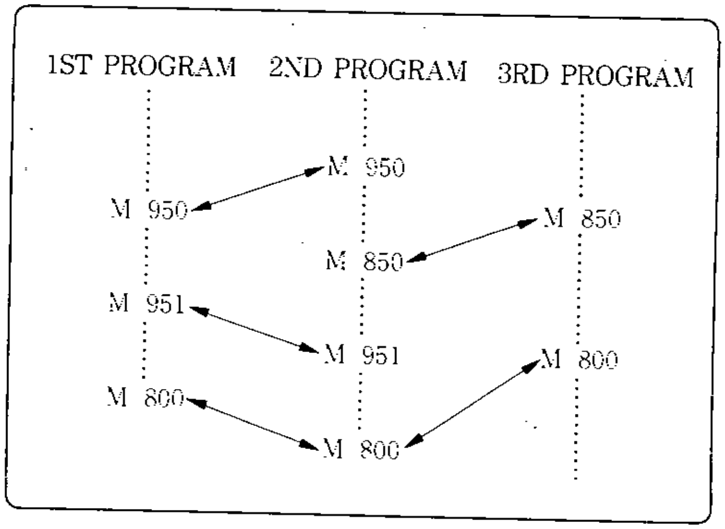


Fig. 2.17.1 Pause M Code Commands

(3) Double synchronization

Double synchronization of programs is made possible by the use of the pause codes listed in **Table 2.17.2**.

Table 2.17.2 Double-pause Commands

Pause Command		Explanation
M code (pm4017 D0 = 0)	G code (pm4017 D0 = 1)	
M998	G998	Double-pause cancel command for 1st and 2nd programs
M999	G999	Double-pause command for 1st and 2nd programs
M948	G978	Double-pause cancel command for 1st and 3rd programs
M949	G979	Double-pause command for 1st and 3rd programs
M898	G958	Double-pause cancel command for 2nd and 3rd programs
M899	G959	Double-pause command for 2nd and 3rd programs
M848	G938	Double-pause cancel command for three programs
M849	G939	Double-pause command for three programs

Double-synchronization is executed as follows. First, synchronize the first double-pause cancel command of one program (M998, for instance) and the double-pause command of another program (M999, for instance.) After the pause command is executed, operation of the other program (where the pause cancel command is executed) is restarted. The program where the pause command is executed enters the pause state. These programs are synchronized again with the cancel command and pause command again, then both programs are restarted.

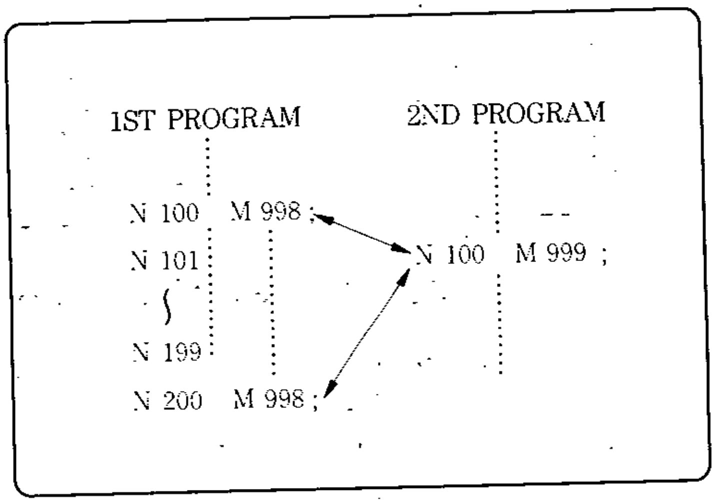


Fig. 2.17.2 Double-pause M Code Commands

In the example shown in **Fig. 2.17.2**, N100 in the 1st program and N100 in the 2nd program are synchronized first. After that, N101 to N199 are executed in the 1st program, during which the 2nd program is suspended. Then N200 of the 1st program and N100 of the 2nd program are synchronized, after which the both programs are restarted.

Use this function to control axes or M commands of different programs.

Fig. 2.17.3 shows double synchronization in three programs.

1ST PROGRAM	2ND PROGRAM	3RD PROGRAM
N 1000 M 848 ;		
N 1001 ;	N 2000 M 849 ;	
		N 3000 M 849 ;
N 1099 ;		
N 1100 M 848 ;		

① All programs are synchronized at N1000, N2000, and N3000.

② N1000 to N1099 are executed in the 1st program.

③ All programs are synchronized at N1100, N2000, and N3000.

④ All programs are restarted.

(a) Pause command used in two programs

1ST PROGRAM	2ND PROGRAM	3RD PROGRAM
	N 2000 M 848 ;	
N 1000 M 848 ;	N 2001 ;	N 3000 M 849 ;
N 1001 ;		
	N 2099 ;	
N 1099 ;	N 2100 M 848 ;	
N 1100 M 848 ;		

① All programs are synchronized at N1000, N2000, and N3000.

② N1000 to N1099 are executed in the 1st program.
N2000 to N2099 are executed in the 2nd program.

③ All programs are synchronized at N1100, N2100, and N3000.

④ All programs are restarted.

(b) Pause command used in one program

Fig. 2.17.3 Double Synchronization in Three Programs

(4) M code used for synchronizing programs

If parameter pm4027 D0 is 0 to use M codes for synchronizing programs, M codes are processed internally, and no output occurs. If parameter pm4027 D0 is 1 to use G codes for synchronizing programs, M codes are processed as normally.

(5) Synchronization-disregard input

To disregard program synchronization when checking the program, use the following synchronization-disregard input parameters:

#3045 D0:—— 1st program synchronization-disregard input

#3046 D0:—— 2nd program synchronization-disregard input

#3047 D0:—— 3rd program synchronization-disregard input

To disregard synchronization with a program, turn on the corresponding input. It is assumed that synchronization with that program has been completed.

This function takes effect under different conditions, depending on the machine. Refer to the manual issued by the manufacturer of the milling machine.

NOTES

1. **Reset during pause**

   If the equipment is reset while completion of a certain program is waited for, the pause status of the reset program is cleared.

2. **Single-block command**

   If program synchronization is to be performed, do not use the synchronization command code, regardless of whether M or G code is used, for specifying sequence numbers.

2.17.2 Password Command ✻

Use this function for a multi-program system to put together parts of different programs into a single program so as to select and execute necessary programs for individual parts.

(1) Specifying character for selecting programs

Specify a character that selects programs to be controled by a certain part program.

The controller compares the characters registered by parameters as a password with the specified characters in part programs.

If they match, the subsequent programs are executed or skipped depending on which parts the programs are being executed.

(2) Method of execution

All programs must be in executable status when the first cycle is started. Then, when a password is commanded, the next block is executed if the password is for starting the local program. If the password is for starting another program, the subsequent blocks are skipped until the next password for starting the local program is commanded.

1ST PROGRAM PASSWORD IS HEAD-1.
2ND PROGRAM PASSWORD IS HEAD-2.

```
O 100 ;
G 28 U 0 W 0 ;                RANGE A
G 50 X100. Z200. ;
   :
HEAD-1 ;
T0101 ;                       RANGE B
G99 S1000 M03 ;
   :
HEAD-2 ;
T0202 ;                       RANGE C
   :
M30 ;
HEAD-1 ;
T1111 ;                       RANGE D
   :
M30 ;
```

When this program is executed on the 1st program, ranges A, B, D are executed in this order. When this program is executed on the 2nd program, ranges A and C are executed in this order.

Fig. 2.17.4 Password Specification

(3) Password command parameter

Table 2.17.3 lists parameters where program passwords are to be set. Specify a password with six characters beginning with an alphanumeric character.

Table 2.17.3 Password Specification Parameters

Program	Password Specification Parameters
1st program	pm4180 to pm4185
2nd program	pm4186 to pm4191
3rd program	pm4192 to pm4197

NOTES

1. **Tape loading with a single program number having more than one M02 or M30 code**

   If the password function is to be used, more than one M02 or M30 code is commanded for a single program number. To upload or download such a program, the end of the program must be detected with "%".

   pm3005 D3 = 0 ← "%" is used for the end of tape.
   = 1 ← M02, M03, and M99 are used for the end of tape.

2. **Number of password characters**

   A password must be specified with six characters, no more or no less than that.

3. **Single-block specification**

   The password block must not contain other commands.

2.17.3 Control Axis Select Command *

Use this function in a multi-program system to control axes that belong to another program as local axes.

(1) Control axis select command

Describe control axis select commands in the following format:

G130 X...Z...C...S.. ;

Command following X, Z, C ← Program specification for the subsequent X, Z, and C commands

Command following S ← Spindle specification for the subsequent S command

(2) G131 ;

Use this command to cancel G130.

(3) Local program stop program

To execute the control axis select command, the program that supports the selected axis must be stopped. Use the program synchronization function for this purpose.

The following is an example of programming:

Example of Programming

1ST PROGRAM (X1, Z1)		2ND PROGRAM (X2, Z2)
O100 ;		O200 ;
G28 U0 W0 ;	X1 AND Z1 ARE CONTROLLED.	G28 U0 W0 ;
⋮		⋮
N111 M998 ;	←→	N211 M999 ;
N112 G130 X2 ;	X2 AND Z1 ARE CONTROLLED.	N212 G28 U0 ;
N113 G28 U0 ;		N213 G50 X100. ;
N114 G50 X100. ;		⋮
⋮		M30 ;
N121 G131 ;		
N122 M998 ;		
N123 G28 U0 ;	X1 AND Z1 ARE CONTROLLED.	
N124 G50 X100.		
⋮		
M30 ;		

Fig. 2.17.5 Control Axis Select Command

(4) Setting coordinate system

Different axes are controlled at the same address before and after a control axis select command on the X-, Z-, or C-axis. Therefore, the coordinate system that has been used is lost upon the command.

Therefore, place the coordinate setup command to set up a new coordinate system after G131 or G130, or after synchronizing programs after control of an axis is changed.

NOTES

1. **G130... ; G131 ; command**

   This command must not be described with other commands in a single block.

2. **Precedence of control axis select command over local section spindle select command**

   The spindle and the rotating tool of the local program can be changed by command G132, as explained described in Par. 2.17.4.

   Command G130 can be used in the G132 mode.
   Command G132 cannot be used in the G130 mode. If used, alarm 0341 occurs.

3. **Independence of control axis select command from synchronous feed reference spindle select command**

   The synchronous feed reference spindle can be changed by command G136, but G136 does not change the spindle command output, as explained in Par. 2.17.5.

4. **Reset in G130 mode**

   If the equipment is reset in the G130 mode, the G131 mode is entered and configuration of axes before command G130 is restored.

5. **Spindle-combined C-axis**

   For a spindle-combined C-axis, the same program must be selected for the spindle and the C-axis.

2.17.4 Local Section Spindle Select Function *

Selects which spindle is used for the spindle command (S command) when two spindles, the mian spindle and rotating tool spindle, are provided in one program.

(1) G132: Command

S command is used for rotation tool spindle command after changing the local section spindle command to the rotating tool spindle side.

S command can be provided for G132 block.

(2) G133: Command

S command is used for the main spindle command after changing the rotating tool spindle to the main spindle.

Example of Programming

```
O100 ;
G28 U0 W0 ;
  :
G132 S1000 ;  ┐
  :           │ Milling with
  :           │ rotating tool
G133 ;        ┘
S500 ;
  :
M30 ;
```

Fig. 2.17.5 Local Program Spindle Select Command

NOTES

1. Singel-block specification
   The G132 or G133 block must not contain other commands except S command.
2. Unusuable for 3rd program
   Command G132 is unusuable for the 3rd program.
3. Unusuable is G130 mode
   Command G132 is unusuable in the G130 mode.
4. After the equipment is reset, the G133 mode is entered.

2.17.5 Synchronous Feed Reference Spindle Select *

A multi-program system may have more than one reference spindle for synchronous feed (mm/rev. feed.) Usually, the single reference spindle is determined for each program and needs not be changed. This function is provided in case the reference spindle needs to be changed for synchronous feed of axes of different programs using the control axis select command.

(1) G136S... ;

This command changes the synchronous feed reference spindle from the initial status. The number after S indicates the spindle number to be used as the reference of synchronous feed after the command.

Table 2.17.4 Synchronous Feed Reference Spindle Initial Status

Pogram	Synchronous Feed Reference Spindle Initial Status
1st program	S1
2nd program	S2
3rd program	S1

Example of Programming

```
1ST PROGRAM
O100 ;
G28 U 0  W 0 ;
 :
G136 S 2 ;
G01
 :
 M30 ;
```

Fig. 2.17.7 Synchronous Feed Reference Spindle Selection

(2) G137 ;

This command cancels synchronous feed reference spindle selection and restores the initial status.

NOTES

1. **Independence of spindle selection by G136 from S command output**
   The spindle selected by command G136 only changes the reference for synchronous feed, but does not change the S command output. Consequently, even after G136S2; has been commanded from the 1st section, S1000; takes effect on S1.
2. **Specifying S of other program in G136**
   No interlock occurs because of double pause at another program.
3. **After he equipment is reset, the G137 mode is entered.**

2.17.6 Remote Program M, T Command *

Use this function to output command M or T from one program to another in a multi-program system.

(1) | G134 M....T...P... ; |

Use the above command to output command M or T from one program to another. M or T is the code to be output. P is the destination program number.

If P is omitted in command G134 output in the 1st program, it is regarded as the command destined to the 2nd program.

If such a command is output in the 2nd program, the 1st program is regarded as the destination.. Command P cannot be omitted if the command is used in the 3rd program.

(2) Interface between controller and machine

Programming restrictions are different in the two types of interfaces that may be used. Refer to the manual issued by the manufacturer of the milling machine to see which interface is used.

Table 2.17.5 Interface for M, T Command to Remote Program

Parameter	Interface	I/O Channel	Output Example (from 1st to 2nd Program)
pm5012 D7 = 0	Interface A	Local section output channel	#3534 D0 to D7 (MF, TF) #3520 to #3529 (M code)
pm5012 D7 = 1	Interface B	Remote section output channel	#3565 D0 to D7 (MF, TF) #3550 to #3559 (M code)

(3) Interface and status of destination program

If interface A is used, command G134 can be output regardless of the status of the destination program.

If interface B is used, G134 for command M can be output only when the destination program is in the pause or double-pause status. For command T, G134 can be output only when the destination program is in the double-pause status. If the destination program is not in the specified status, alarm 0340 occurs.

NOTES

1. **M command to remote program**

   M00, M01, M02, M30, internal M code, preread stop M code, and macro call M code cannot be output.

2. **T command to remote program**

   Offset numbers cannot be specified.

2.17.7 Pulse Copy Function *

Use this function to output pulses to one axis in synchronization to another axis in a multi-program system.

(1) Pulse copy start

Pulse copy is started when the signal is input to the controller. Specify the source (#3086, #3088) and destination (#3087, #3089) for each axis.

Two sets of pulses can be copied at the same time. The sign of pulses can be inverted by parameter (pm5003 D6, D7.)

(2) Pulse copy with destination and source axes in the same program

If the destination axis is not moving, pulses can be copied. If the destination and the source axes are in different programs, pulse copy can be commanded in the auto mode only when the destination axis is stopped or in the double-pause state. (Alarm "2196")

(3) Pulse copy end

Pulse copy can be ended only when the source axis is stopped.

(4) Setting up coordinate system after pulse copy

After pulse copy, the coordinate system of the destination axis is lost. Set up the coordinate system again after pulse copy has been completed.

Example of Programming

```
1ST PROGRAM                                   2ND PROGRAM

O100 ;                                        O200 ;
G28 U0 W0                                     G28 U0 W0 .
    :                                             :
N101 M998 ;                                       :
M * * * ;  ←— PULSE COPY ON                   N201 M999 ;
    :                                             :
M * * * ;  ←— PULSE COPY OFF                      :
M998 ;                                        M30 ;
    :
M30 ;
```

Fig. 2.17.7 Program Example of Copying Pulses from 1st Program Axis to 2nd Program

NOTE

During pulse copy, neither the source nor destination axis can be returned to the reference point at low speed. If it is attempted, alarm 2196 occurs.

SECTION 3 DESCRIPTION OF OPERATION

This section outlines operations of the NC operator panel and 9-inch CRT display, key functions, and screen display.

NOTE For the explanation of NC operator panel with 14" CRT display, refer to the manual issued by the manufacturer of the milling machine.

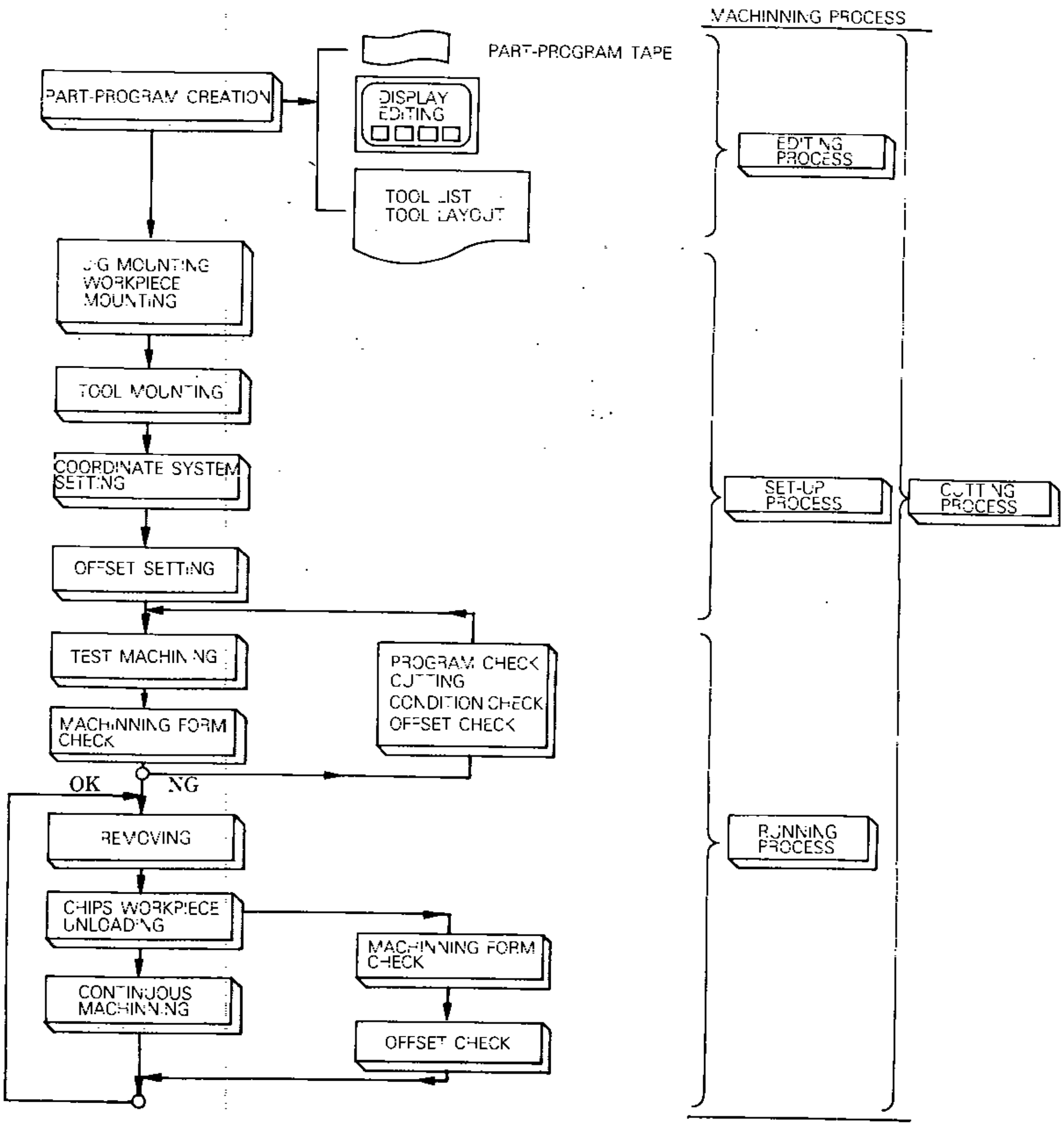
SECTION 3 CONTENTS

	Page
3 DESCRIPTION OF OPERATION	347
3.1 OUTLINE OF SCREENS	350
3.2 CONFIGURATION OF PROCESSES, JOBS, AND FUNCTIONS	351
3.3 NC OPERATOR FOR PANEL AND DISPLAY SCREEN	354
3.3.1 Outline of Operator Panel Key Functions	354
3.3.1.1 Power on/off buttons	354
3.3.1.2 9-inch graphic display	355
3.3.1.3 Process keys	356
3.3.1.4 Soft keys	356
3.3.1.5 Address keys	358
3.3.1.6 Data keys	359
3.3.1.7 Page keys	359
3.3.1.8 Cursor keys	360
3.3.1.9 Action keys	361
3.3.1.10 Reset key	362
3.3.1.11 AUX keys	362
3.4 DISPLAY AND WRITE	363
3.4.1 Constant Display	363
3.4.2 Pop-up Menu	368
3.4.3 Key Buffer Editing Function	369
3.4.4 Buzzer Function	370
3.4.5 PROG (Program Editing) Process	371
3.4.5.1 Part program editing	372
3.4.5.2 Part program directory job (job soft key [DIR])	395
3.4.5.3 Part program I/O verification ([IN/OUT] job soft key)	411
3.4.6 SET Setup Process	439
3.4.6.1 Offset function	440
3.4.6.2 Work coordinate system shift function	445
3.4.7 Run Process RUN	447
3.4.7.1 Program job	448
3.4.7.2 Command value job (COMMAND VALUE job soft key)	463
3.4.7.3 Setting job	464
3.4.8 Maintenance Process MAINT	471
3.4.8.1 Parameter job	472
3.4.8.2 I/O monitor job	479
3.4.8.3 I/O verification job	485
3.4.9 Common Process COMN	491
3.4.9.1 Absolute position job (POS job soft key)	492
3.4.9.2 Alarm job (ALARM job soft key)	500
3.4.9.3 Time job (TIME job soft key)	504
3.4.10 Power-on Initial Indication	509

3

MEMO

The man-machine interface YASNAC i series is designed based on complete understanding of "**what is cutting**". The configuration of the operator panel (keyboard) and screen display provides ease of operation.



Machinning Processes

3.1 OUTLINE OF SCREENS

The display screen consists of five blocks of process screens: Program editing, set-up, running, maintenance, and common process screens.

- **Workpiece cutting process screens**
  ..... Program editing process screen
  ..... Set-up process screen
  ..... Running process screen
- **Controller maintenance process screen**
  ..... Maintenance process screen
- **Screen used in common in the above four processes**
  ..... Common process screen

Display control keys (three cutting process keys, a maintenance process key, and a common process key) are assigned to the five screens. These keys facilitate quick access to necessary screens. This is one feature of the human-oriented NC.

3.2 CONFIGURATION OF PROCESSES, JOBS, AND FUNCTIONS

Each process has jobs, and each of the jobs has functions. Up to five jobs can be assigned to a process.

NOTE The number of functions for a job is not limited. Use the soft key to select jobs and functions.

Fig. 3.2.1 shows and example of process structure (Soft key names are indicated in brackets):

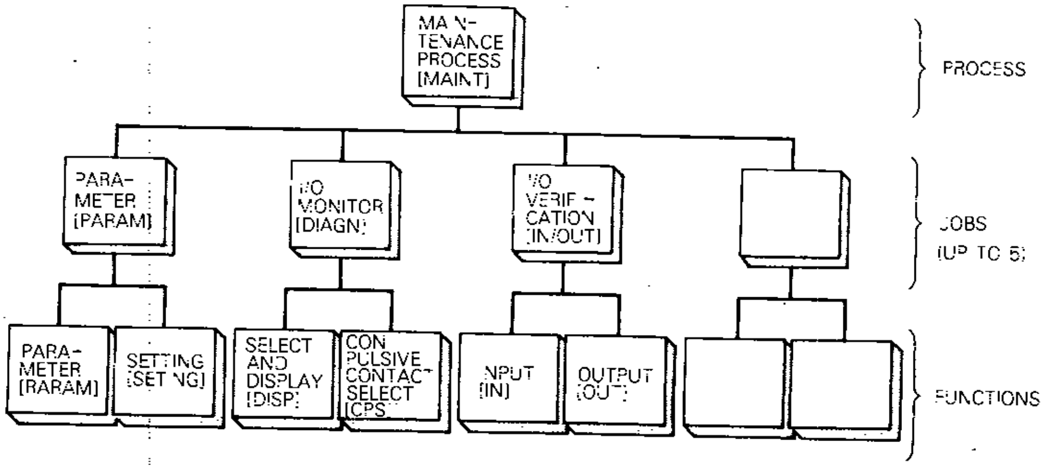
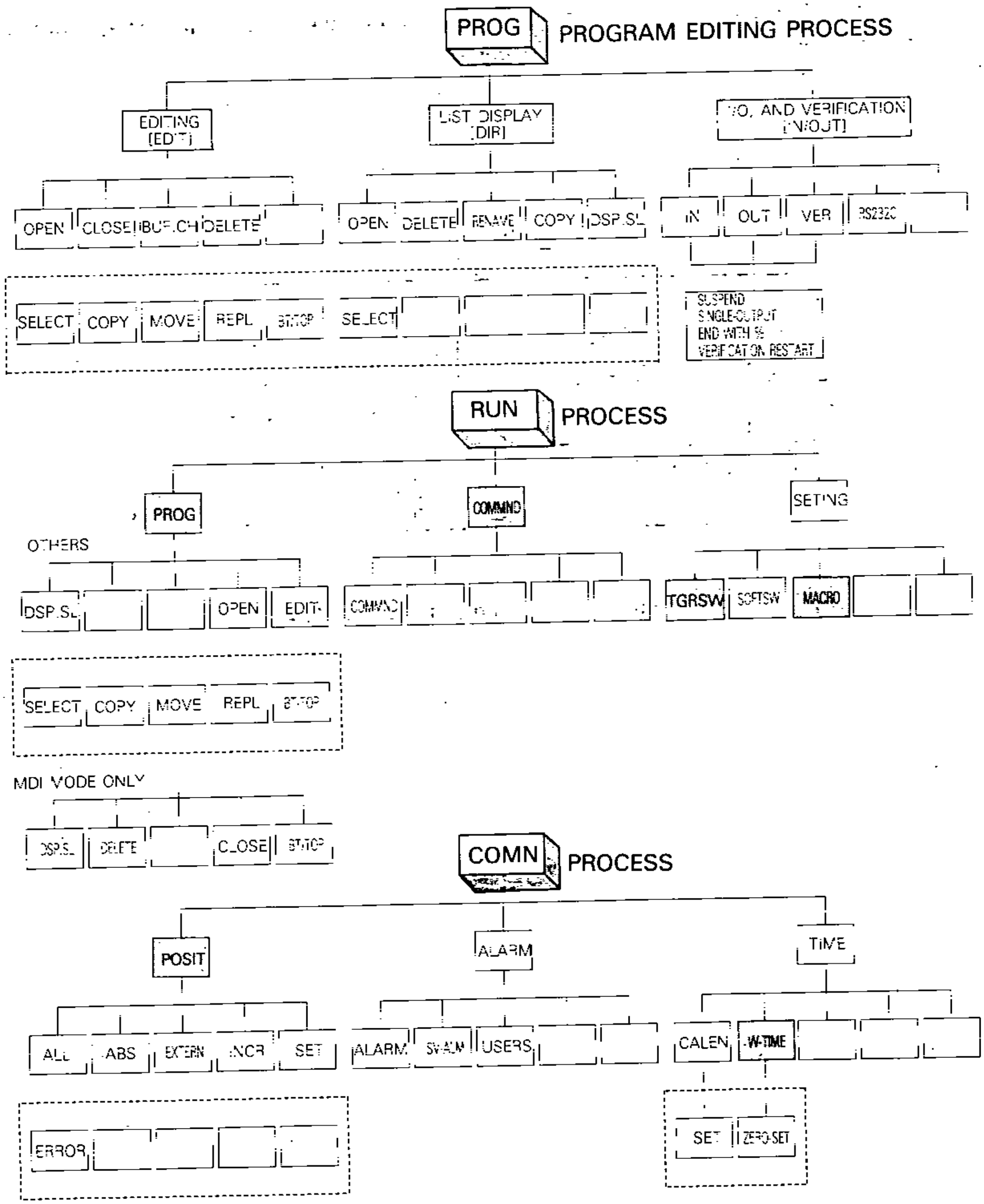
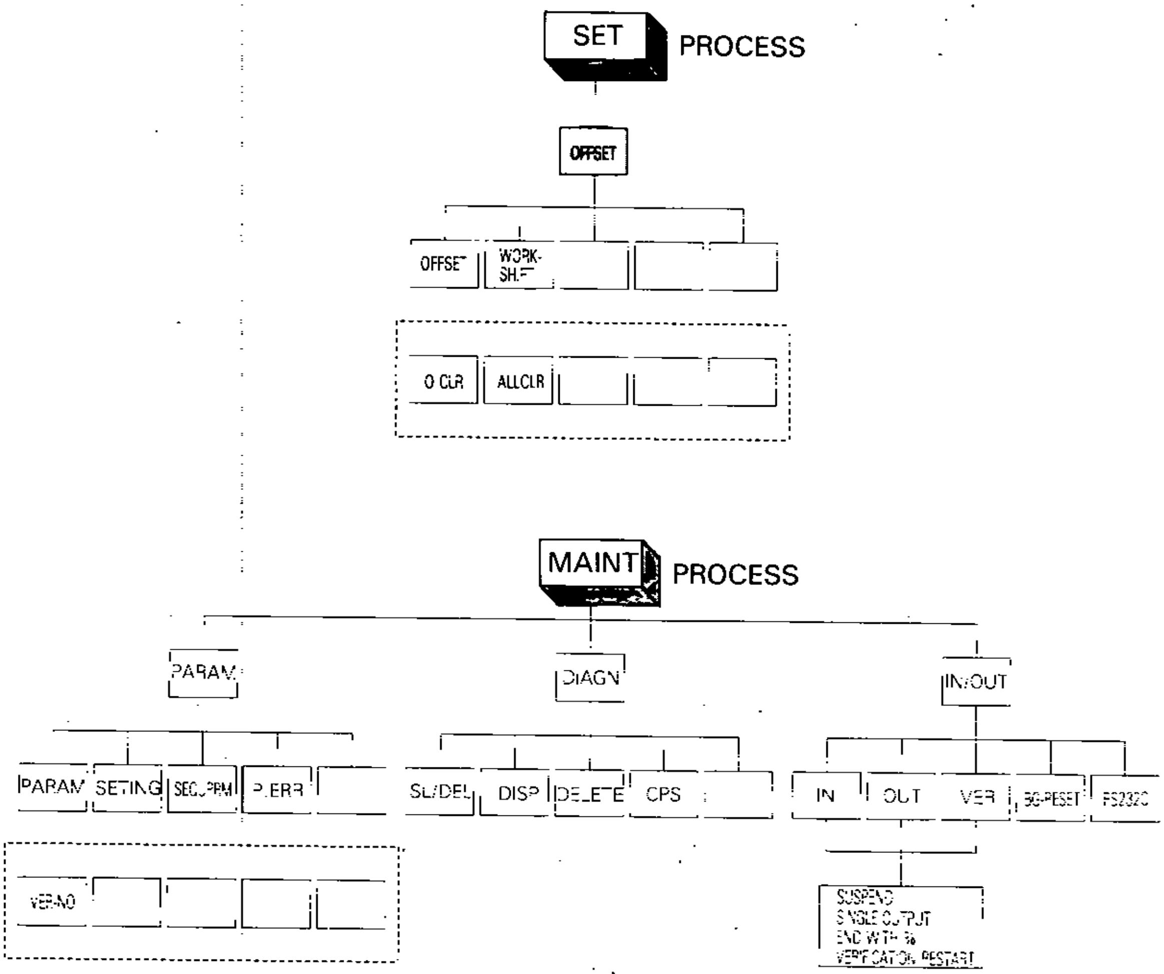


Fig. 3.2.1 Example of Process Structure

■ YASNAC i80L OPERATION TREE

The following is a tree diagram of YASNAC i80L operations (Soft key names are indicated in brackets):





: DISPLAYED FOR EACH PROGRAM

: DISPLAYED BY THE FUNCTION DISPLAY CHANGE KEY ( )

3.3 NC OPERATOR PANEL AND DISPLAY SCREEN

3.3.1 Outline of Operator Panel Key Functions

Fig. 3.1.1 shows the operator panel of YASNAC i series:

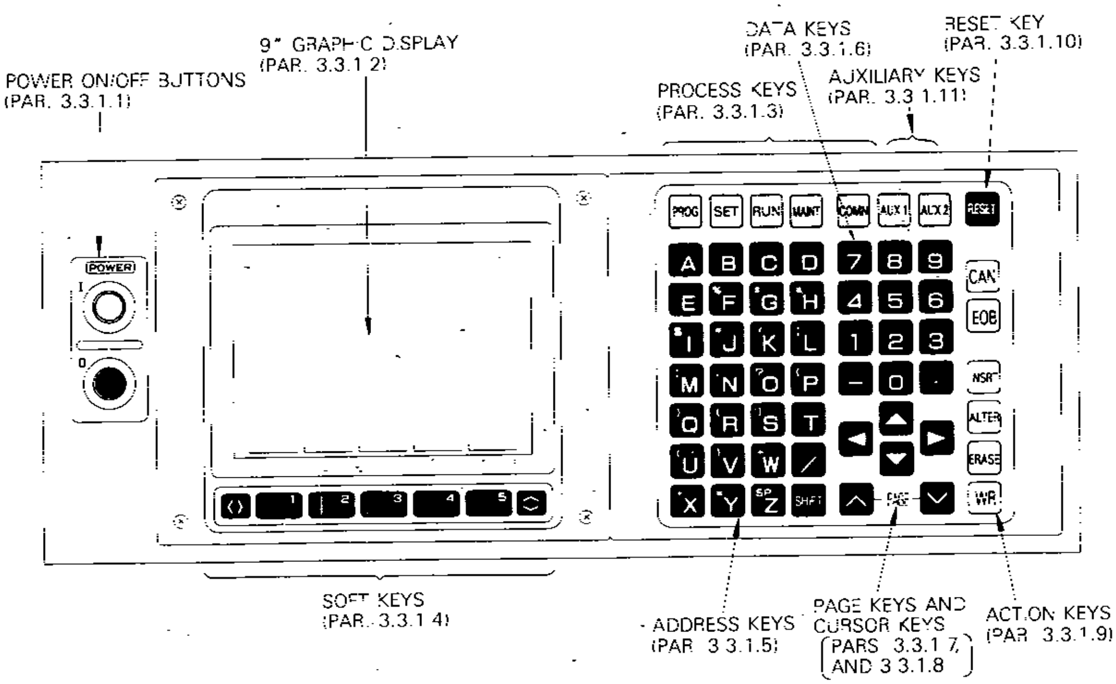
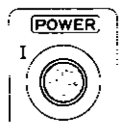


Fig. 3.3.1 NC Operator Panel

3.3.1.1 Power on/off buttons



Power-on button

This button is for turning on power to the controller.

If the button is depressed once, only controller power is turned on. If the button is depressed twice, servo power is also turned on.

NOTE With some machines, servo power may be turned on even if the button is depressed only once. Refer to the manual by the milling machine manufacturer.

Power-off button

This button is used to turn off power to the controller.

Controller and servo power are turned off at the same time when this button is depressed.

3.1.1.2 9-inch graphic display

The graphic screen displays data in alphanumeric characters of 1 × 1 to 3 × 3 sizes.

Maximum Number of Characters	40 characters × 20 lines (or 25 lines) = 800 characters (or 1000 characters) [1 × 1 size]
Display Characters	Numbers (0 to 9, −, .) Alphanumeric characters (A to Z) Special codes [/ (slash,) EOB, +, #, SP, =, etc.] Katakana and JIS level-1 kanji set } (1 × 1 size only)
Graphic Display	640 − 400 dots (high precision type)

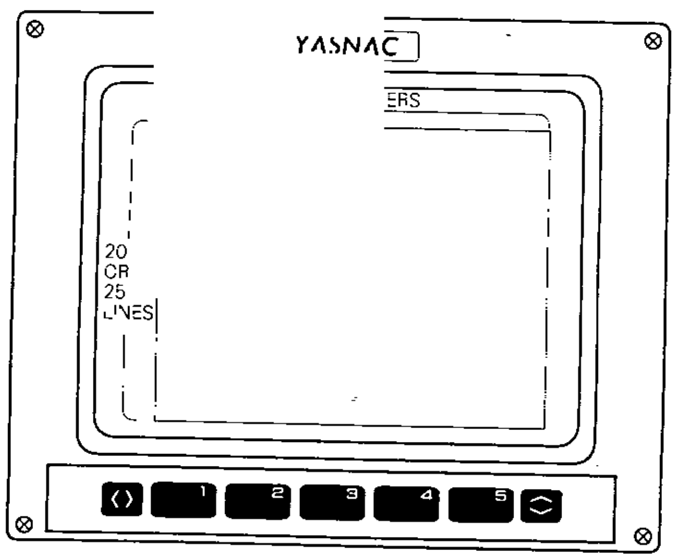


Fig. 3.3.2 9-inch Graphic Display

3.3.1.3 Process keys (PROG SET RUN MAINT COMN)

Process keys are select keys used for dispaly and writing.

There are five process keys. Depressing one of the keys enters the corresponding process.

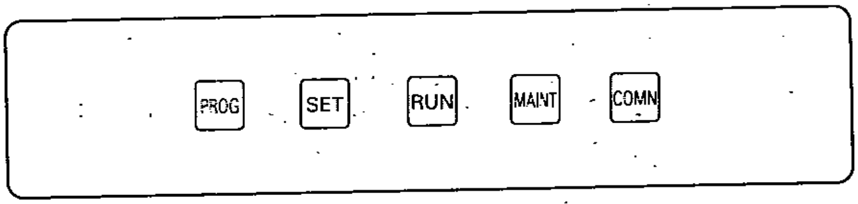


Fig. 3.3.3 Process Key

PROG	Program editing key:	Selects the program editing process.
SET	Set-up key :	Selects the set-up process.
RUN	Running key :	Selects the running process.
MAINT	Maintenance key :	Selects the maintenance process.
COMN	Common key :	Selects the common process.

3.3.1.4 Soft keys ⟨⟩ 1 2 3 4 5 ⇕

Soft keys are select keys used for display and writing.

NOTE In the following explanation, the term "soft keys" refers to the menu frames dispalyed on the lower part of the screen, although actual keys to be depressed are from 1 to 5 corresponding to the five frames. This is for easy understanding of key functions.

There are two lines of soft keys, as shown in **Fig. 3.3.4.**

The upper line is called the job soft keys. The lower line is called the function soft keys.

Depress ⇕ to enter the job select mode or the function select mode.

The current mode can be determined by the soft keys or the up/down indicator on the screen.

Select jobs in the job select mode and select functions in the function select mode.

In the function select mode, if six or more functions are available, function select soft keys on the next page are displayed by depressing ⟨⟩.

NOTE If six or more functions are available, the function display change indicator is displayed as shown in Fig. 3.3.4.

Depress soft keys 1 to 5 under the above conditions to select jobs or functions. The selected job or function is highlighted.

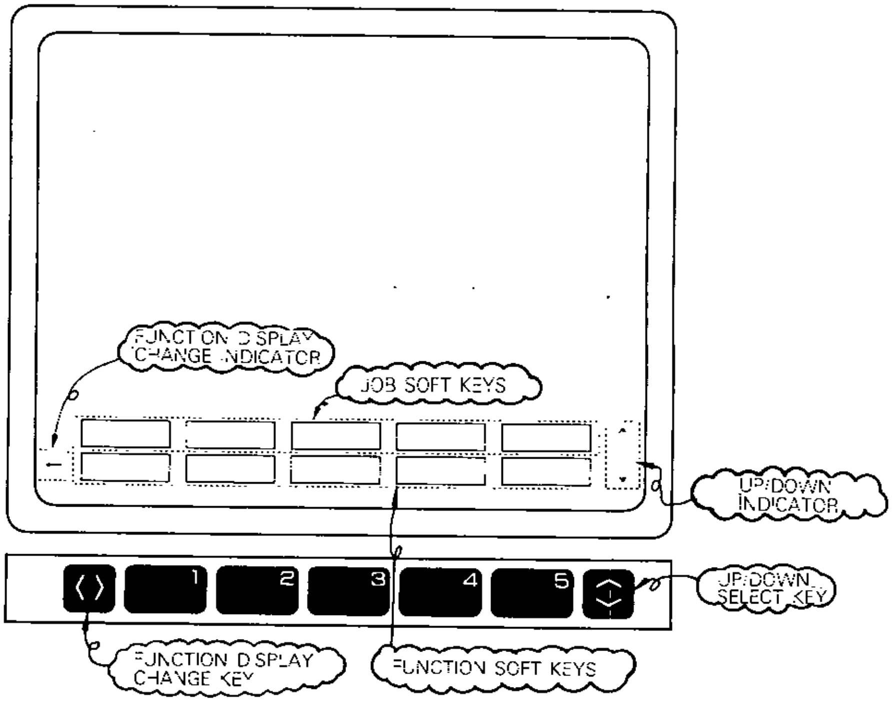


Fig. 3.3.4 Soft Keys

NOTE Move UP/DOWN indicator, and the box shifts upward or downward.
With top position, job select mode is displayed, for bottom position, function select mode.

3.3.1.5 Address keys (A to Z / SHIFT)

Address keys are used to specify address characters to write data.

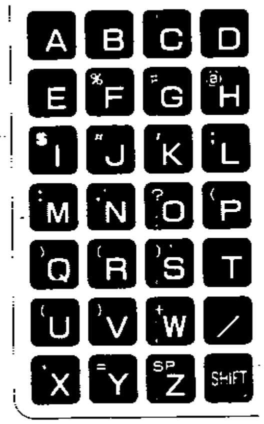


Fig. 3.3.5 Address Keys (28 keys)

⟨Meanings of special characters⟩

/ : Use for optional block skip command.

SHIFT : Depress the shift key before depressing F to Z to enter the special character indicated on the upper left of each key.

NOTE The special characters are used for macro program operators.

3.3.1.6 Data keys (0 to 9 − .)

Data keys are used to write all numerical data including program command values, tool offset, setting, and parameters.

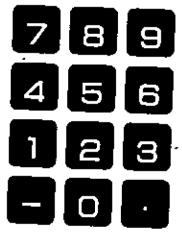


Fig. 3.3.6 Data Keys (12 Keys)

0 to 9 : Use to enter numbers.

− : Use to enter sign.

. : Use to enter decimal point.

3.3.1.7 Page keys (∧ and ∨)

Page keys are used to dispaly the next or preceding screen called "page".

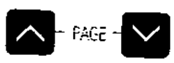


Fig. 3.3.7 Page Keys

(1) To display the next page, depress ∨.

(2) To display the preceding page, depress ∧.

(3) Holding down either of the above keys, pages are turned successively.

3.3.1.8 Cursor keys ()

Cursor keys are used to move the cursor on the screen.
The cursor is a highlighted indicator on the screen.

(1) Up and down cursor keys ( and )

(a) Depress  to move the cursor toward the beginning.

(b) Depress  to move the cursor toward the end.

(c) Holding down either of the above keys, the cursor moves backward or forward continuously.

(d) To move the cursor to a specific number, enter the number and depress key  or 

(2) Left and right cursor keys 

(a) Depress  to move the cursor to the left.

(b) Depress  to move the cursor to the right.

3.3.1.9 Action keys (CAN EOB INSRT ALTER ERASE WR)

Action keys are used for various purposes. When an action key is depressed, the screen is changed or data are input. Action keys change the display.

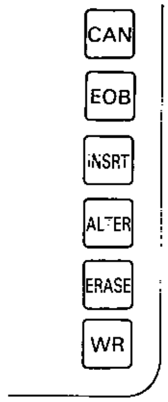


Fig. 3.3.8 Action Keys (6 Keys)

(cancel) CAN : Used to cancel wrong numerical or address data.

(end of block) EOB : Used to specify the end of a block.
On the screen, semicolon (;) is displayed for EOB.

(insert) INSRT : Used to insert data in memory.

(alter) ALTER : Used to alter data in memory.

(erase) ERASE : Used to erase data in memory.

(write) WR : Used to write data that have been entered by address and data keys.

3.3.1.10 Reset key ([RESET])

The reset key resets the internal status of the controller:

(1) When the [RESET] key is depressed, the following operations are executed:

- Move commands are canceled.
- The preread buffer is cleared.
- Alarms are cleared if the cause has been removed.
- Tool offset functions are canceled.
- Auxiliary functions are canceled.
- The label skip function is turned on.
- Memory is rewound.
- The reset signal is sent off.
- G codes are reset.
- The key buffer is cleared.

(2) The following items are not changed by reset:

- Absolute position of each axis
- F command
- S and T commands
- Tool offset data, setting data, and parameter data

3.3.1.11 AUX keys ([AUX1] [AUX2])

Use these keys to extend option functions.

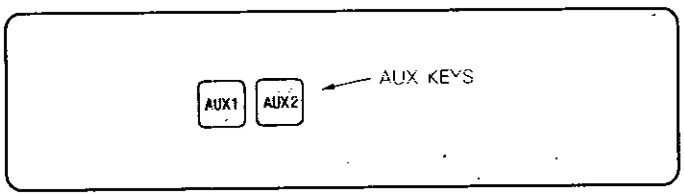


Fig. 3.3.9 AUX Keys

[AUX1] (Auxiliary key 1) : Used to switch programs. The key is disregarded in a system where a program cannot be changed.

[AUX2] (Auxiliary key 2) : Used to add option functions.

3.4 DISPLAY AND WRITE

3.4.1 Constant Display

Regardless of which process, job, or function is selected, constant display is output at the top and the last six lines on the screen, as shown **Fig. 3.4.1**:

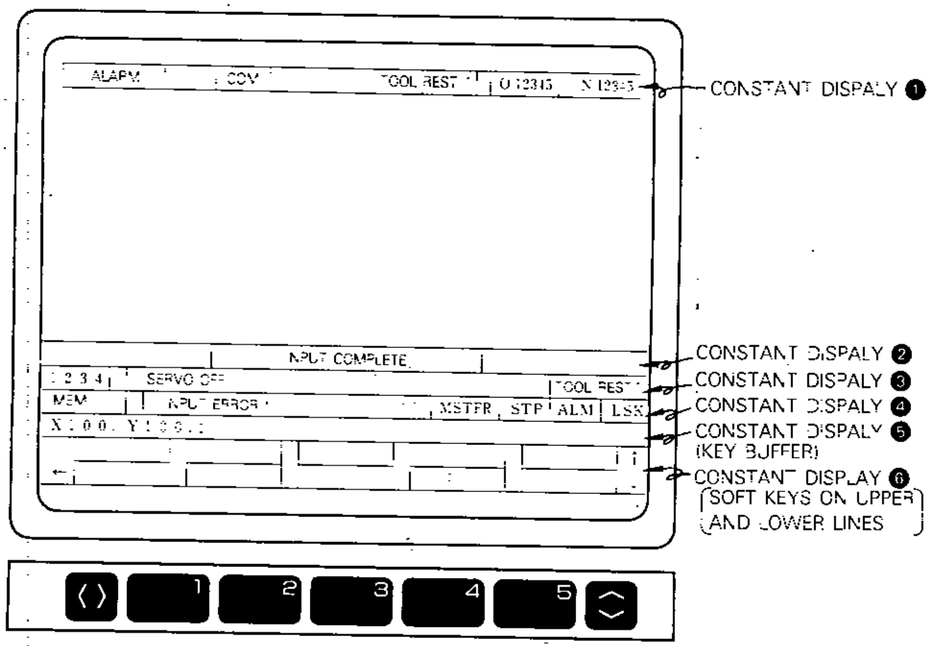


Fig. 3.4.1 Constant Display

Table 3.4.1 explain the constant display items:

Table 3.4.1 Constant Dispaly Items No.

No.	Item	Message	Explanation
❶	Screen title	ALARM, OFFSET, ⋮ etc.	Selected job name of the screen is displayed as screen title.
	Process indication	EDT, SET, RUN, MNT, COM	Selected process name is displayed.
	Program indication	Tool rest 1 Tool rest 2 Tool rest 3	Selected program name is displayed. This indication is not displayed for single-program systems. Seven patterns can be selected by parameter pm3105.

Table 3.4.1 Constant Display Items (Cont'd)

No.	Item	Message	Explanation
	Program no.	O ＊＊＊＊＊ O 0 0 0 0 1 to O 9 9 9 9 9	Program number of running program in selected program is displayed. "O＊＊＊＊＊" is displayed if no program number has been selected.
	Sequence no.	N 0 0 0 0 0 to N 9 9 9 9 9	Sequence number of running sequence in selected program is displayed.
❷	Operation results and prompting message	INPUT OK? (Y/N) INPUT INPUT COMPLETE OUTPUT OK? (Y/N) OUTPUT OUTPUT COMPLETE VERIFY OK? (Y/N) VERIFY VERIFY COMPLETE CREATE COMPLETE COPY OK? (Y/N) COPY COMPLETE RENAME OK? (Y/N) RENAME COMPLETE DELETE OK? (Y/N) DELETE COMPLETE SEARCH COMPLETE INPUT O NO. INPUT COMMENT. CLOSE COMPLETE CLEAR ALL OFFSET ? (Y/N) INPUT O NO. CLOSING SEACHING MODE IS UNSTABL	Results of key operations and status of execution of equipment are displayed.

Table 3.4.1 Constant Display Items (Cont'd)

No.	Item	Message	Explanation
3	Alarm no.	0 0 0 1 to 9 9 9 9	Top priority alarm in selected progrm is dispaiyed. If no alarm occurs in selected program, top priority alarm in other programs is dispalyed.
	Alarm message	TH ERROR ILLEGAL CHARACTER EXTERNAL ⋮ (etc.)	Message about top priority alarm in selected program is displayed. If no alarm occurs in selected program, message about top priority alarm in other programs is displayed.
	Alarm location	Tool rest 1 Tool rest 2 Tool rest 3	When alarm message is displyed, the alarm program is displayed.
4	Operation mode	EDIT MEM MDI TAPE STEP HANDL JOG RAPID	Mode of operation of selected program is displayed.

Table 3.4.1 Constant Display Items (Cont'd)

No.	Item	Message	Explanation
4	Warning	INPUT ERROR! NOT FOUND O NO! NOT FOUND! ALREADY IN! EACH MEM. OVER! PROGRAM OVER! VERIFY ERROR! MACRO LOCK! LINE LOCK! RUNNING PROGRAM! NC RUNNING! FORMAT ERROR! EDIT LOCK! NOT FOUND! ALREADY EDIT! SELECT MODE ERR! COPY MODE ERROR! MOVE MODE ERROR! INPUT ERROR! SETTING PRM ERR! SET PROHIBIT! MDI BUF OVER! UNSUITABLE PRM! PROGRAM READING! FIX DATA READING! PROGRAM EDITING!	Warning is a less serious alarm that does not operation. To clear warning, enter from keyboard, change mode, or change screen.

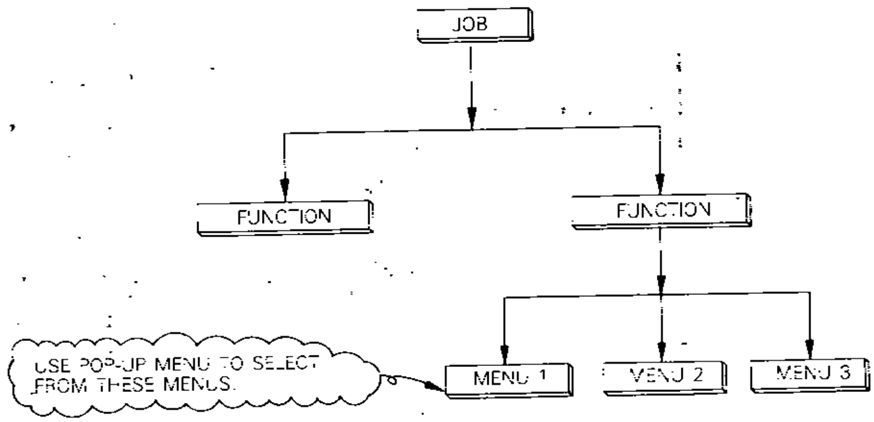
Table 3.4.1 Constant Display Items (Cont'd)

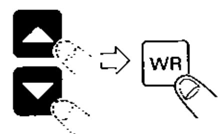
No.	Item	Message	Explanation
4	Operation status	M/S/T/F/R DWELL	Status of selected program is displayed. M, S, and T indicates that completion of corresponding code is waited. F, R, and DWELL indicates cutting, rapid traverse, and dwell is being executed, respectively.
	Operation status		Status of selected program is displayed.
		STP	Selected program is stopped.
		RST	Selected program is reset.
		BUF	Selected program is preread.
	Alarm status		Status of selected program is displayed.
		ALM	Alarm occurred.
		BAT	Battery alarm occurred.
		A/B	Both alarm and battery alarm occurred.
		BGA	Background alarm occurred.
		B/B	Both background alarm and battery alarm occurred.
		EBA	Absolute battery alarm occured.
		E/B	Both Absolute battery alarm and background alarm occurred.
		B/E	Both battery alarm and absolute battery alarm occurred.
		A/E	Both alarm and absolute battery alarm occurred.
	Label skip	LSK	Displayed when label skip is on. Programs are distinguished.
5	Key buffer		Data entered from keyboard is echoed back. (Up to 40 characters can be input.)
6	Soft keys		Select keys about display and write. (Upper and lower lines)

3.4.2 Pop-up Menu

Use the pop-up menu to select a menu in a function.

Usually, one function has one menu. The pop-up menu is displayed when a choice of two or more menus is provided.



(Example) When the [DELETE] soft key is depressed, the pop-up menu appears as shown in the figure. The menu disappears when the operator has selected one. ()

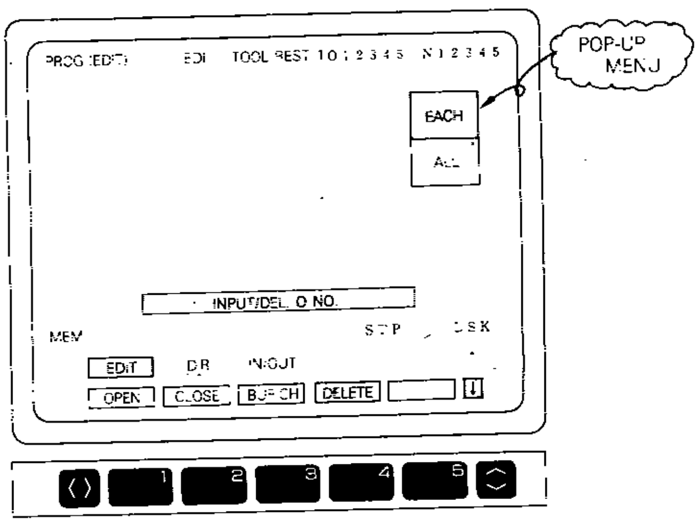


Fig. 3.4.2 Pop-up Menu for [DELETE]

3.4.3 Key Buffer Editing Function

Characters can be inserted in or deleted from data entered in the key buffer display area on the screen.

Use this function to correct errors when,

- An error is found immediately after you typed data, or
- Warning message "INPUT ERROR!" appears after you depressed the WR key.

(1) The key buffer cursor indicated as an underline in this manual blinks on the first column of the key buffer area. As characters are entered, the cursor moves to the left of the string.

(2) If cursor key ◀ or ▶ is depressed after characters have been entered in the key buffer area, the key buffer cursor moves accordingly. In other words, the cursor key takes effect in the key buffer area only.

NOTE The cursor keys move the cursor in the whole screen when no data have been input in the key buffer area.

(3) Editing in the key buffer area is explained in the following examples:

(a) Delete

```
G00 X100. Z100. ; _
```

Suppose the above data have been entered and are displayed on the key buffer area. Delete X100.

1. Depress ◀ to move the cursor to "Z".

```
G00 X100. Z100. ;
```

2. Depress CAN the key five times to delete "X100."

```
G00 Z100. ;
```

Each time the CAN key is depressed, the character before the key buffer cursor is deleted.

NOTE If the key buffer cursor is positioned at the first column of the key buffer area, nothing can be deleted.

(b) Insert

| X100. Z100. ; _ |

Suppose the above data have been entered and are dispalyed in the key buffer area. Insert "G00" before "X100."

1. Depress ◄ to move the cursor to "X".

| X100. Z100. ; |

2. Enter G O O The characters are inserted before the key buffer cursor.

| G00 X100. Z100. ; |

3.4.4 Buzzer Function

The buzzer can be sounded when data are entered from the NC operator panel. Whether to sound the buzzer can be set with the parameter as follows:

pm0007 D7 0: Not sound the operator panel buzzer.
1: Sound the operator panel buzzer.

3.4.5 [PROG] (Program Editing) Process

Part programs are created in the editing process.

The following operations are executed in the process: part program editing, program list display, and I/O with external equipment.

These operations can be started anytime regardless of the controller operation mode. The operations can also be started during automatic operation.

The operations are classified into the following three jobs: part program editing, part program list display, and part program I/O verification.

Procedures for the operations are explained in the following:

1 Depress the [PROG] key. Any of the following jobs is displayed.

• Part progrm editing	See Par. 3.4.5.1 (page 372.)
• Part program directly job	See Par. 3.4.5.2 (page 395.)
• Part program I/O verification	See Par. 3.4.5.3 (page 411.)

2 To display a necessary job, depress the corresponding job soft key ([EDIT], [DIR], or [IN/OUT].) Otherwise, depress the [PROG] key

to switch the displayed job.

3.4.5.1 Part program editing

Depress the [EDIT] job soft key. The following functions are available in

this job:

(1) Part program call (page 373) : Calls a part program to be edited on the screen.

(2) Part program edit (page 376) : Edits the part program called on the screen.

(3) Part program save (page 390): Saves the part program that was called to edit on the screen.

(4) Part program delete (page 392) Deletes a part program.

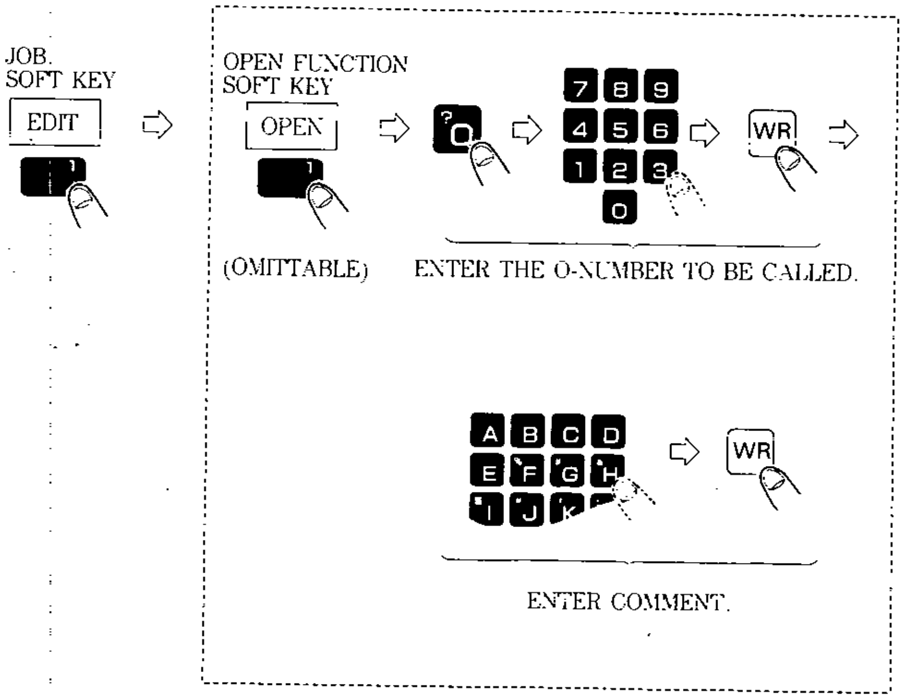
Details of the above functions are described in the following.

Up to three programs can be edited at the same time using an option.

(1) Calling part program to edit.

Call a part program to be edited on the screen. If the specified O-number is not in memory, a new program is created.

KEYPOINT



⇨ With the use of an option, up to three part programs can be called at the same time by repeating the operation in the above box.

The [OPEN] function soft key need not be depressed.

NOTE
1. In the editing prohibited status, new program numbers cannot be created. Clear the editing prohibited status in advance.
2. A currently running program can also be called, but cannot be edited.

1 Depress the [EDIT] job soft key.

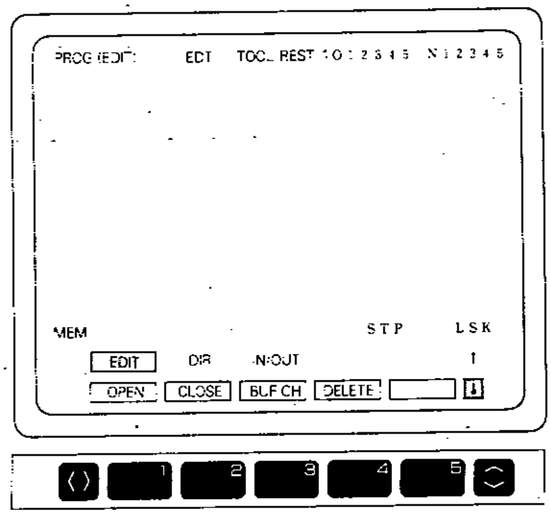


Fig. 3.4.3 Part Program Call Screen

2 Depress the [OPEN] function soft key. 

"INPUT O NO." is displayed. This step can be skipped.

3 Enter the O-number of the program to call.

(For example )

4 If the input O-number is in memory, the contents of the part program are displayed. If the number was not found, "INPUT COMMENT" is displayed.

5 Enter comment.

(For example )

To skip comment, depress the  key only. If the specified O-number was not found, the new number is entered in memory and displayed on the screen.

The contents of the called part program are displayed as shown in **Fig. 3.4.4**:

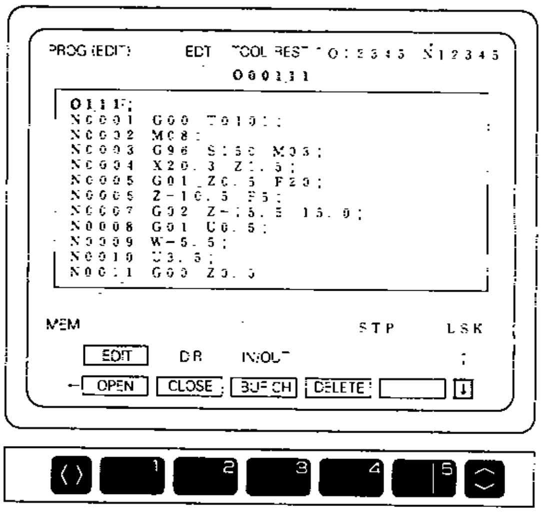


Fig. 3.4.4 Contents of (One) Called Part Program

6 If another program is to be called, depress the [OPEN] function soft key.

"INPUT O NO." is displayed. Up to three part prgorams can be called at the same time using an option.

Fig. 3.4.5 is the screen display of two part programs. **Fig. 3.4.6** is the screen display of three part programs.

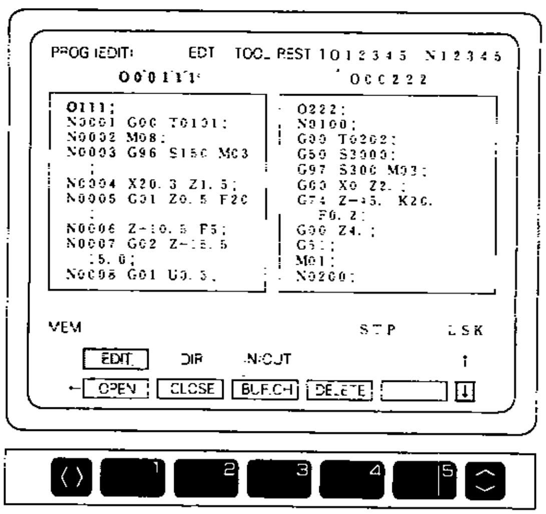


Fig. 3.4.5 Contents of (Two) Called Part Programs

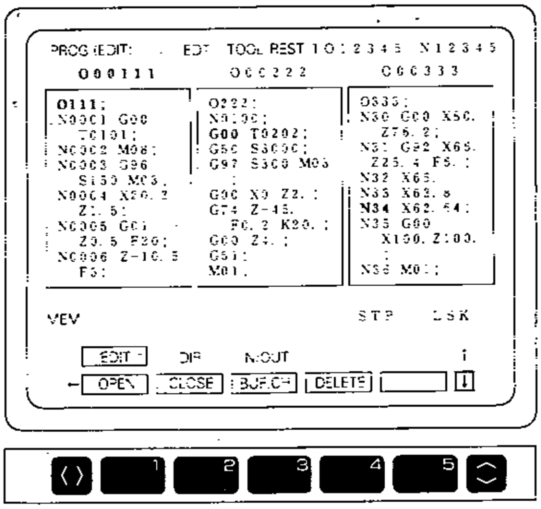


Fig. 3.4.6 Contents of (Three) Called Part Programs

NOTE

1. If a program is called although there already is a maximum number of programs that can be edited at the same time, the new program is opened after a program that was selected by the [BUF.CH] function soft key has been automatically saved.
   3 : [OPEN]
2. A program can be called without using the function soft key: 1
   - Enter the O-number of the program to call, then depress the WR key.
   - If the O-number is in memory, the program is called.
   - If the O-number is not in memory, "INPUT COMMENT" is displayed, indicating that the program is new.
   - Enter comment and depress the WR key.

(2) Editing part program

Edit the part programs called on the screen.

Expansion editing functions (copy, move, alter/erase with range specified) can be used with an option.

KEYPOINT

SOFT FUNCTION KEY

BUF.CH ⇨ [SELECT] [COPY] [MOVE] [REPL] [BT/TOP]

INSRT ALTER ERASE

3

MOVE THE CURSOR TO THE O-NUMBER OF THE PROGRAM TO BE EDITED.

USE THE ABOVE KEYS TO EDIT THE PROGRAM.

1 If more than one program has been called, depress the [BUF.CH] function soft key for necessary number of times to move the cursor to the O-number of the program to be edited.

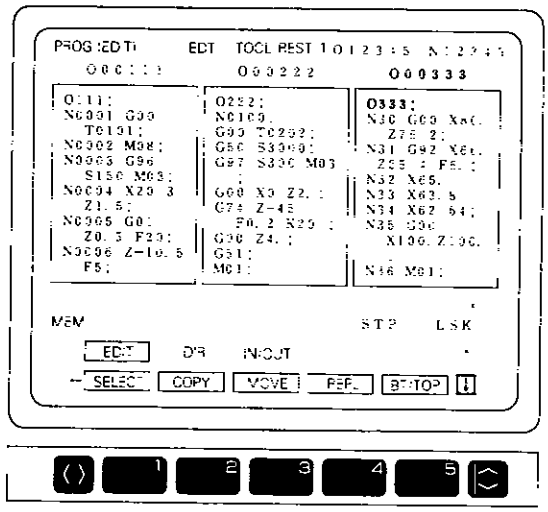


Fig. 3.4.7 Part Program Editing Screen

2 Edit programs using the following keys: function soft keys ([SELECT], [COPY], [MOVE], [REPL], [BT/TOP]) editing keys cursor keys INSRT ALTER ERASE, and page keys .

Functions of these keys are explained in the following.

KEYPOINT

INSRT	:	Inserts data immediately after the word indicated by the cursor. **[See (a) on page 379.]**
ALTER	:	Replaces the word indicated by the cursor with new data. **[See (b) on page 380.]**
ERASE	:	Erases the word indicated by the cursor. **[See (c) on page 382.]**
◀ ▲ ▼ ▶	:	Move the cursor to the preceding or next block. Enable address search. **[See (d) and (e) on page 383.]**
∧ PAGE ∨	:	Display the preceding or next screen.

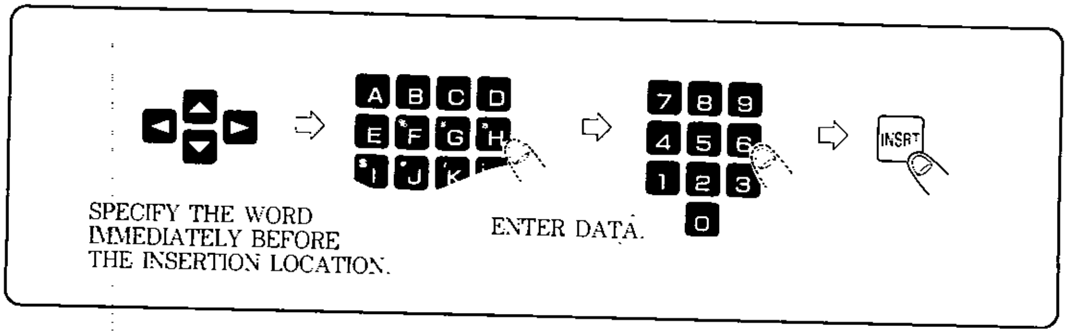
Function soft keys

[SELECT]	:	Specifies the beginning of a character string to [COPY] or [MOVE].	
[COPY]	:	Copies data. **[See (f) on page 384.]**	Inserts a copy of an already registered character string in another place. Maximum character string length: 1024 characters
[MOVE]	:	Moves data. **[See (g) on page 386.]**	Moves an already registered character string to another place. Maximum character string length: 1024 characters
[REPL]	:	Replaces data. **[See (h) on page 388.]**	Searches for a specified character string and replaces it with a new one.
[BT/TOP]	:	Moves the cursor to the last address of the program displayed on the screen. If depressed again, moves the cursor to the first address of the program displayed on the screen. The destinations are switched every time the key is depressed.	

NOTE
1. In the editing prohibited status, editing is impossible. Clear the editing prohibited status in advance.
2. A currently running program cannot be edited. Stop the program before editing.

(a) Inserting word

KEYPOINT



Place the cursor at the word immediately before the location where new data are to be inserted. Enter the data, then depress the  key. The data are inserted immediately after the word indicated by the cursor.

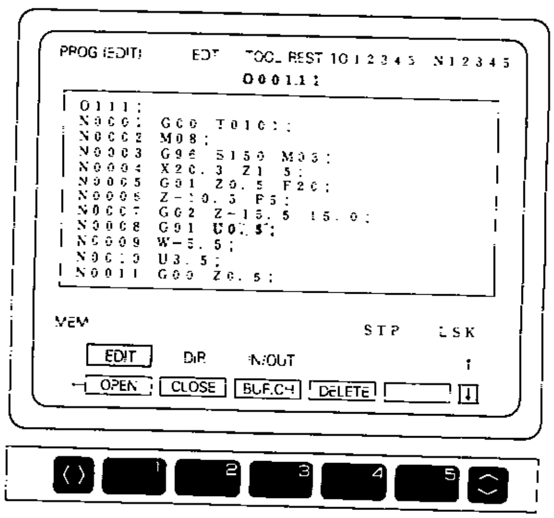


Fig. 3.4.8 Screen before Insertion

For example, enter 

The screen changes as follows:

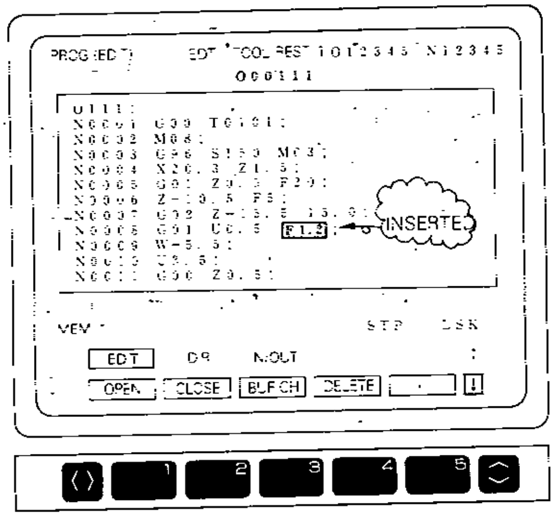
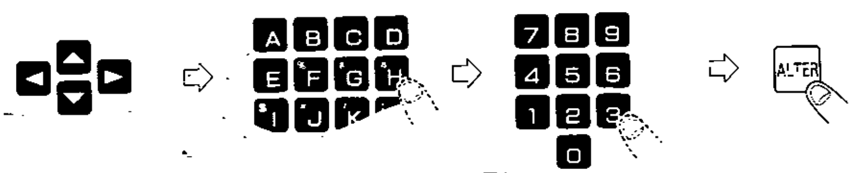


Fig. 3.4.9 Screen after Insertion

NOTE After insertion, the cursor moves to the last input word.
Up to 40 words can be inserted at one time by this operation.

(b) Altering word 

KEYPOINT



SPECIFY THE WORD TO BE CHANGED.

ENTER DATA.

Place the cursor at the word to be changed, enter new data, then depress the  key.

The word at the cursor location is erased and replaced by the input data.

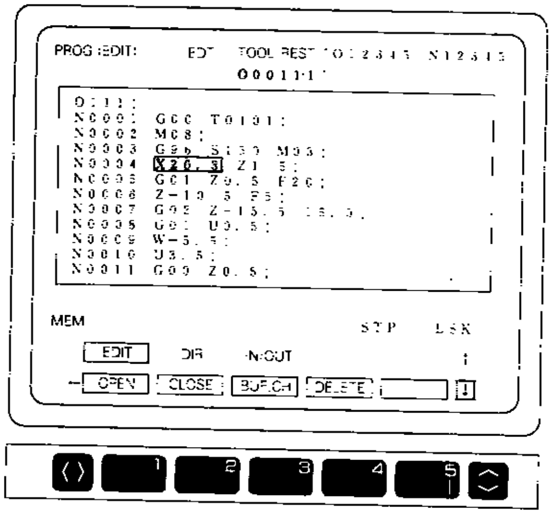


Fig. 3.4.10 Screen before Change

For example, enter 

.

The screen changes as follows:

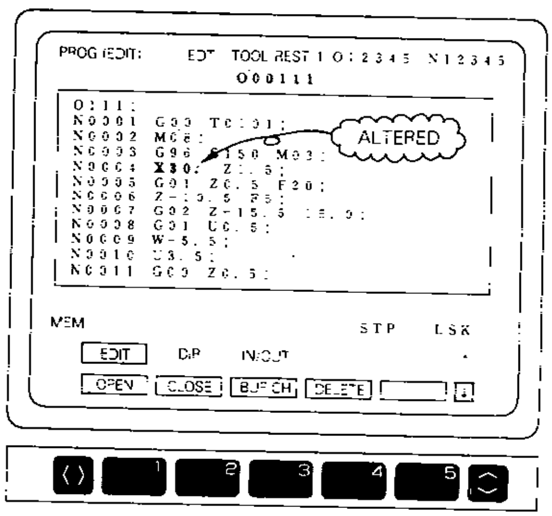
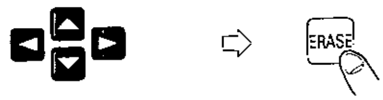


Fig. 3.4.11 Screen after Change

NOTE If more than one word (up to 40 characters) is entered, only one word is replaced by the input data.

(c) Erasing word ERASE

KEYPOINT



MOVE THE CURSOR TO
THE WORD TO BE ERASED.

Place the cursor at the word to be erased, and depress the ERASE key. Only that word is erased.

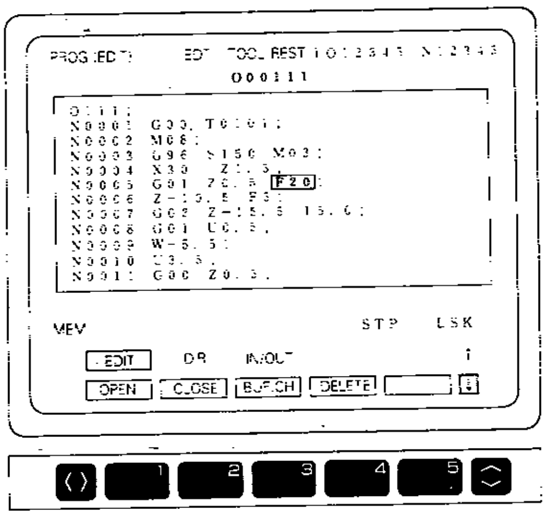


Fig. 3.4.12 Screen before Erasure

Depress ERASE the key. The screen changes as follows:

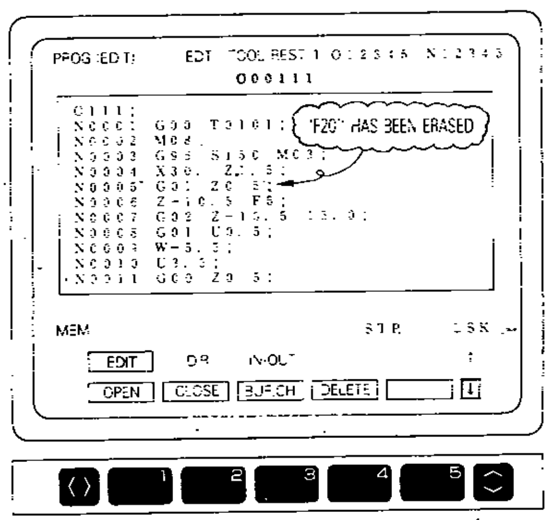


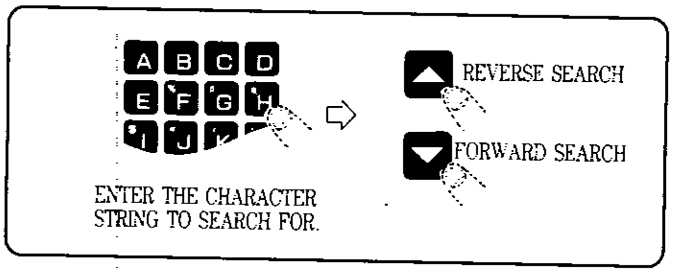
Fig. 3.4.13 Screen after Erasure

(d) Moving cursor

1. Depress the up and down cursor keys and to move the cursor to the preceding or next block in a part program.
   The cursor is always at the beginning of a block.
2. Depress the left and right cursor keys and to move the cursor to the preceding or next word.
3. Depress the page keys and to turn pages of a part program.

(e) Address search

KEYPOINT



1. Enter the character string to search for.
2. Depress cursor key or

- Depress cursor key to search from the current cursor location to the end of the program. During the search, "SEARCHING" is displayed.
- Depress cursor key to search from the current cursor location toward the beginning of the program.

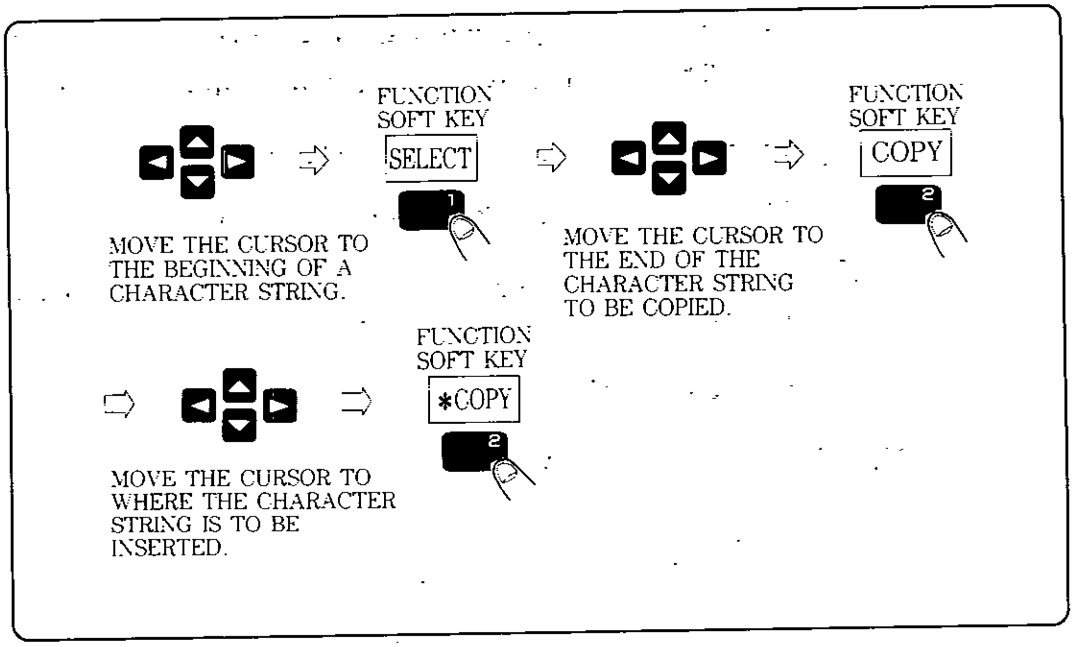
3. If the target word is found, the message disappears and the cursor moves to that word.
4. If the target word was not found, a warning message is issued.
5. To repeat search for the same character string toward the end of the program, depress SHIFT and at the same time. To repeat toward the beginning of the program, depress SHIFT and at the same time.

   Search operation can be repeated any number of times until a different character string is searched for.

NOTE In address search, the exact typed character string is searched for. Do not omit leading zeros. (Pattern search function)

(f) Copy ([COPY] function soft key)

KEYPOINT



❶ Move the cursor to the beginning of a character string.

❷ Depress the [SELECT] function soft key.

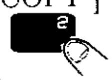
Indication of the key "SELECT" is highlighted and the select mode is entered.

NOTE If the [SELECT] function soft key is depressed when it is highlighted, the select mode

is canceled.

❸ Move the cursor to the end of the character string to be copied. The specified character string is highlighted.

❹ Depress the [COPY] function soft key. The selected character



string is saved in memory. At the same time, an asterisk is displayed on the side of the [COPY] function soft key. The character string that has been highlighted returns to normal display.

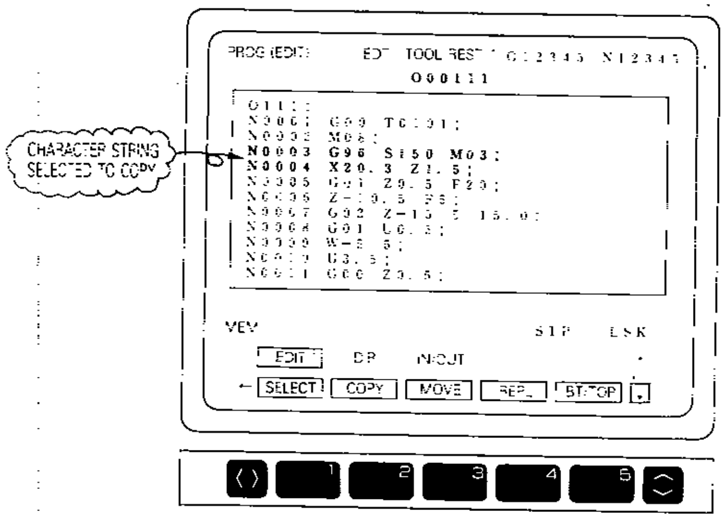


Fig. 3.4.14 Specifying Character String to Copy

❺ Move the cursor to where the character string is to be copied.

❻ Depress the [*COPY] function soft key. The selected character

string is copied at the cursor location.

❼ As long as the asterisk is displayed at the [COPY] function soft key [*COPY], the same character string can be copied over and over

by depressing the function soft key.

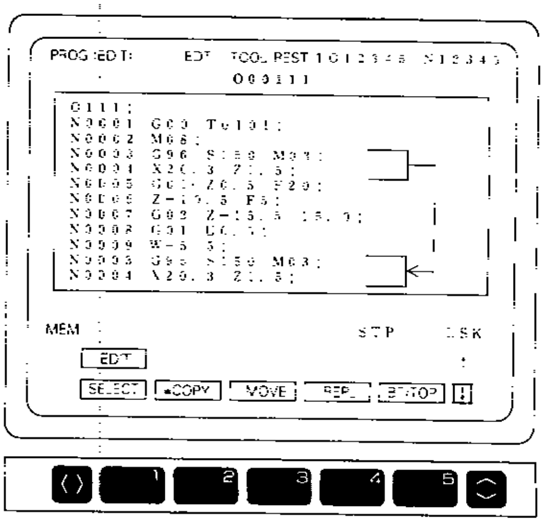


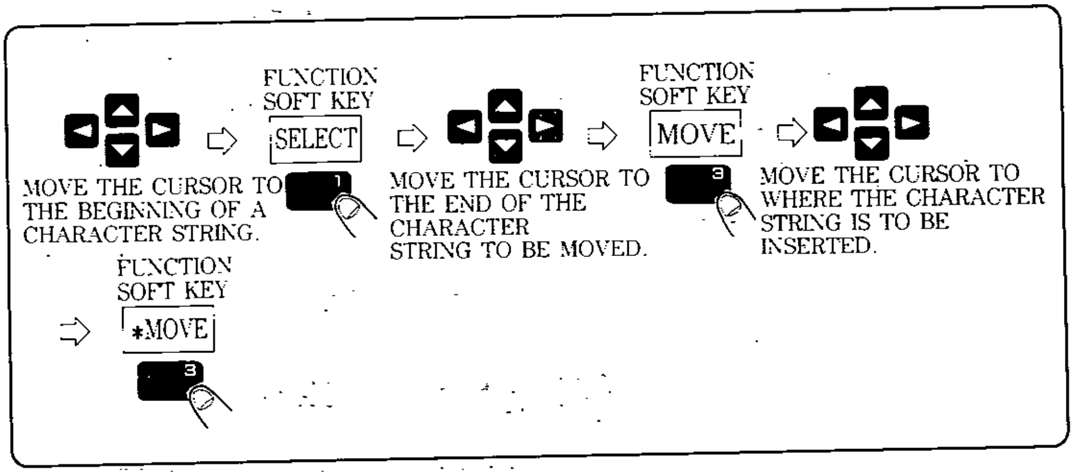
Fig. 3.4.15 Copy Screen

NOTES

1. If power is turned off, the selected character string is cleared and the asterisk disappears.
2. The same character string can be copied in two or more programs without selecting it in each program.
3. If [MOVE] is used after [COPY], the character string selected to copy is cleared.
4. If more than 1024 chracters are selected to copy, "MEM.AREA OVER" is displayed for warning.

(g) Move ([MOVE] function soft key)

KEYPOINT



❶ To specify a character string to be moved, do the same as for copying. The selected character string is highlighted.

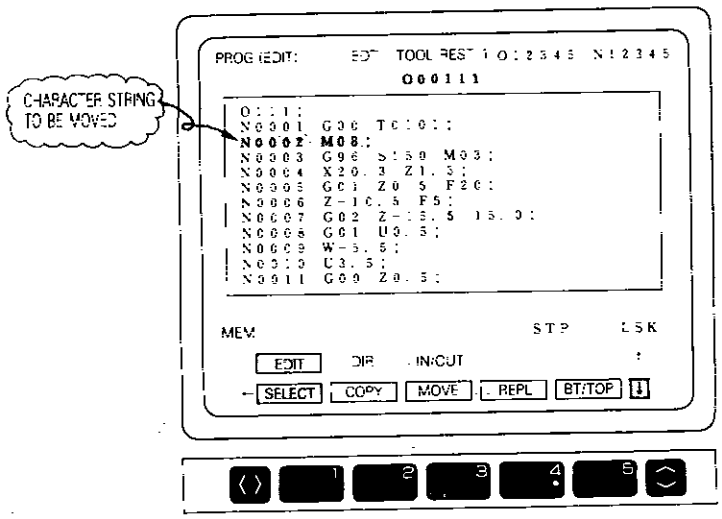
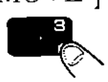


Fig. 3.4.16 Specifying Character String to be Moved

❷ Depress the [MOVE] function soft key. The selected character string



is saved in memory. At the same time, an asterisk is displayed on the side of the function soft key. The character string that has been highlighted is erased.

❸ Move the cursor to where the character string is to be inserted.

❹ Depress the [*MOVE] function soft key. The selected character

string is inserted at the cursor location.

❺ As long as the asterisk is displayed at the [MOVE] function soft key [*MOVE], the same character string can be moved over and over by

depressing the function soft key.

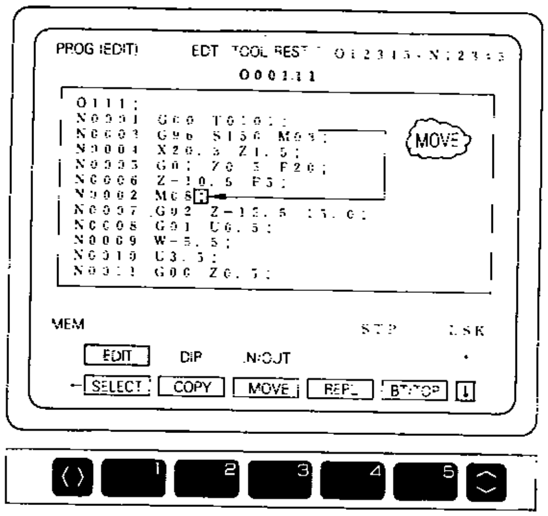
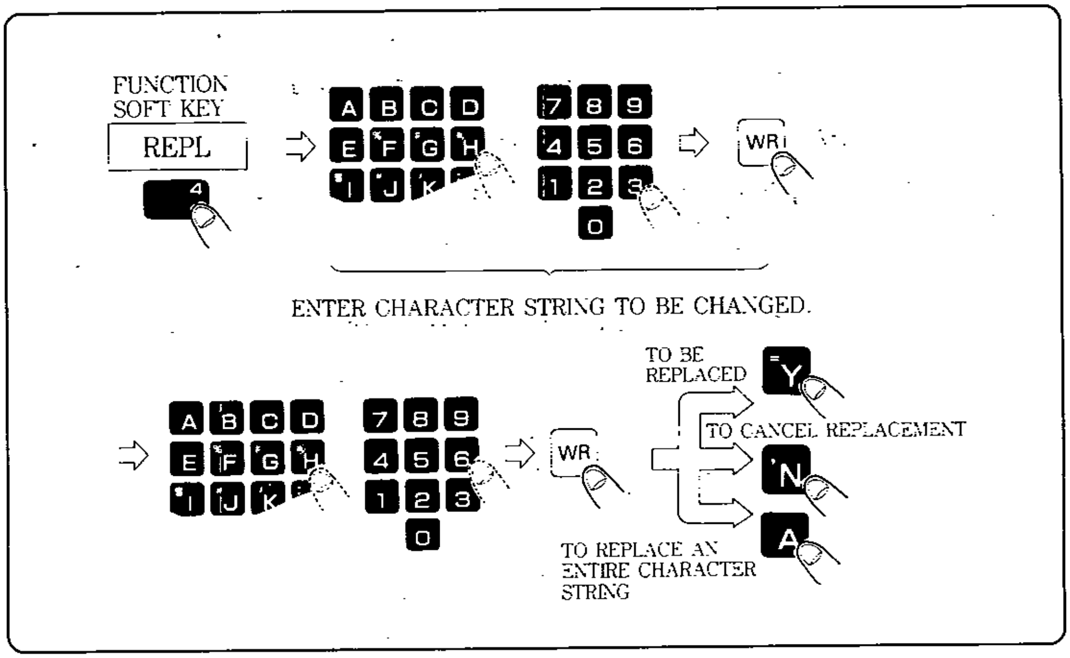


Fig. 3.4.17 Screen after Move

NOTES
1. If power is turned off, the selected character string is cleared and the asterisk disappears.
2. The same character string can be moved through two or more programs.
3. If COPY is used after MOVE, the character string selected to be moved is cleared.
4. If more than 1024 characters are selected to copy, "MEM.AREA OVER" is displayed for warning.
5. If an O-number at the beginning of the program is included, the block cannot be moved. If such is selected, "MOVE MODE ERROR!" is displayed for warning.

(h) Replace ([REPL] function soft key)

KEYPOINT



❶ Depress the [REPL] function soft key. "INPUT OLD STRING" is displayed. Enter the character string to be changed. (For example, to change "G00," enter G 0 0 WR .)

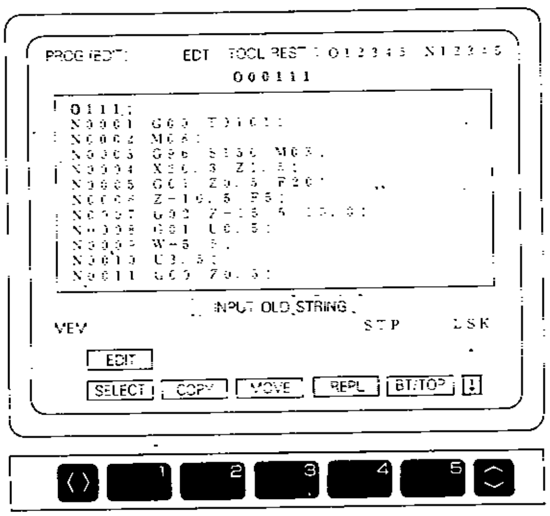


Fig. 3.4.18 Replace Screen

❷ "INPUT NEW STRING" is displayed. Enter a new character string. (For example, to change to "G01," enter G, O, 1, WR.)

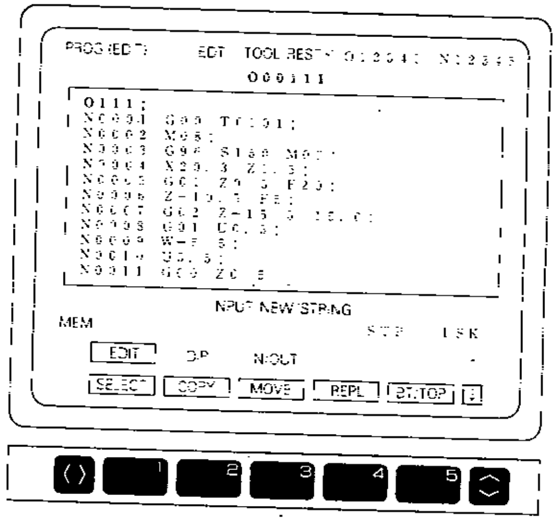


Fig. 3.4.19 New Character String has been Input

❸ The cursor moves to the character string specified in ❶, and "Yes/No/All?" is displayed.

(a) Depress Y to replace the character string specified in ❶ by the one specified in ❷.

If the character string specified in ❶ is at another location, the cursor moves to it.

If the character string specified in ❶ was not found, then the replacement operation ends.

(b) Depress N not to replace the character string. If the character string specified in ❶ is at another location, the cursor moves to it. If the character string specified in ❶ was not found, then the replacement operation ends.

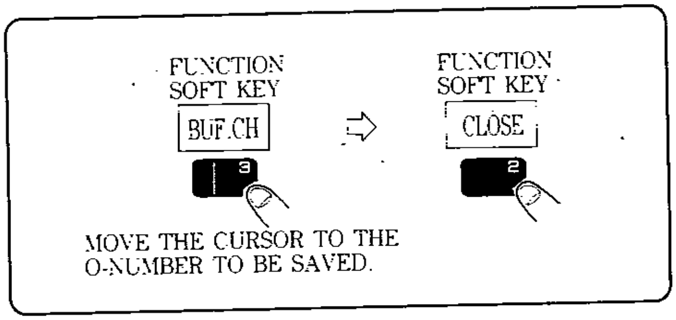
(c) Depress A to replace every character string speified in ❶ by the one specified in ❷, throughout the program displayed on the screen, without asking for approval for each replacement.

(d) To exit from the replacement mode, depress function soft key [REPL] again, or depress the [RESET] key. CAN

(3) Saving part program after editing

Called part programs must be saved after being edited. Part program memory in the controller is rewritten only after the program is stored. After one program is saved, another part program can be called.

KEYPOINT



1. Assume that O-numbers of programs that have been called are displayed on the screen in **Fig. 3.4.20**.

2. Depress the [BUF.CH] function soft key to select the O-number of

the program to be saved. The cursor moves each time the key is depressed.

3. Depress the [CLOSE] function soft key.

NOTE If any process key other than editing process keys is depressed while a part program is being edited, the program is automatically saved temporarily. Depress an editing process key again to continue editing.

Do not turn off power during editing. Save part programs before turning off power.

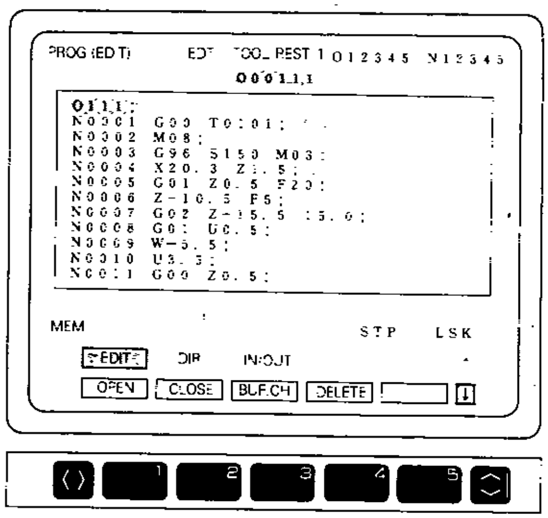


Fig. 3.4.20 Screen before Part Program is Saved

Depress the [CLOSE] function soft key. The screen changes as follows:

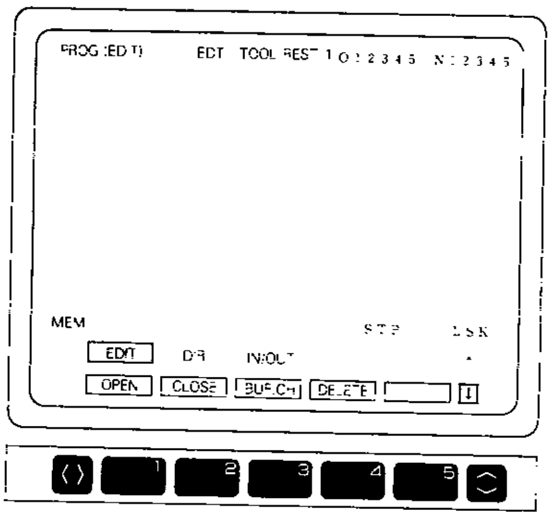


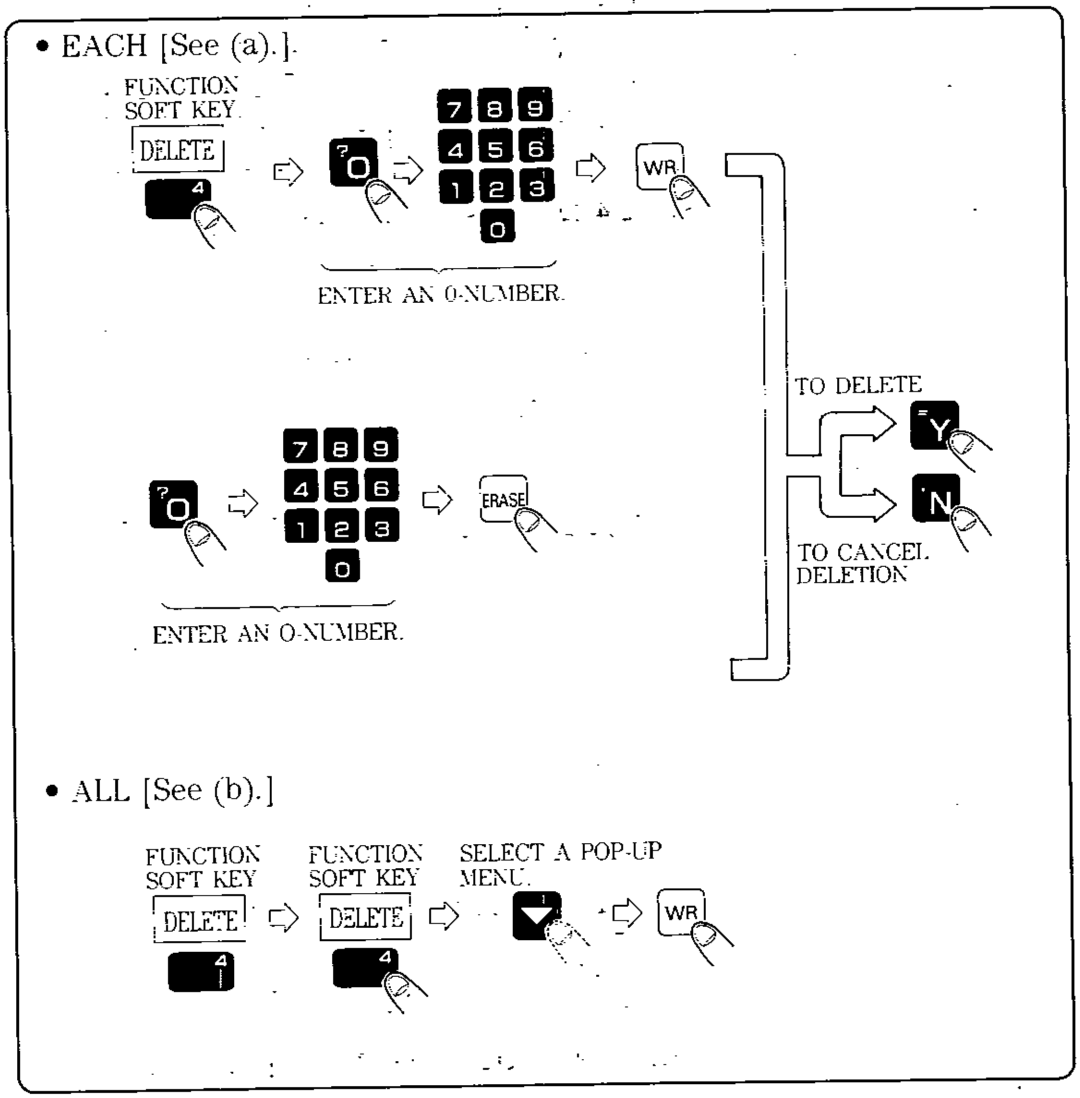
Fig. 3.4.21 Screen after Part Program is Saved

(4) Deleting part program

There are two deleting menus:

EACH	: Delete a single part program.
ALL	: Delete all part programs.

KEYPOINT



NOTE 1. In the editing prohibited status, part programs cannot be deleted. Clear the editing prohibited status in advance.
2. A currently running program cannot be deleted. Stop the program before deleting it.

(a) EACH

There are two procedures:

(i) Using [DELETE] function soft key

[4]

1. Depress the [DELETE] function soft key.

   [4]

   "INPUT DEL O NO." is displayed.

2. Enter the O-number of the program to be deleted.

   (For example, [O] [1] [2] [3] [WR])

(ii) Starting with O-number

1. Enter the O-number of the program to be deleted.

2. Depress [ERASE]

3. After (i) or (ii), "DELETE OK? (Y/N)" is displayed.

   • Depress [Y] to delete the program.

   Then "DELETE COMPLETE" is displayed.

   • Depress [N] not to delete the program.

NOTE If the specified O-number was not found, "NOT FOUND O NO!" is displayed.

(b) ALL

1. Depress the [DELETE] function soft key.

   [4]

2 Depress the [DELETE] function soft key again.

The pop-up menu appears.

3 Depress the cursor key to select "ALL."

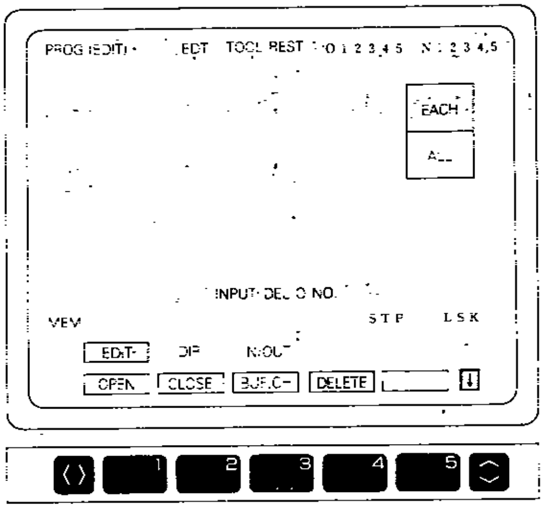


Fig. 3.4.22 Specified Deleting Menu Screen

4 Depress the WR key.

The pop-up menu for selecting the delecting menu disappears. Then "DELETE ALL? (Y/N)" is displayed.

- Depress Y to delete.

  Then "DELETE COMPLETE" is displayed.

- Depress N not to delete the program.

NOTES
1. If the specified O-number is called for editing, the program is automatically saved and then deleted.
2. If the specified O-number is called and running, "O*****" is displayed for the running program, and the status is restored before any program is deleted.

3.4.5.2 Part program directory job (job soft key [DIR])

Depress the [DIR] job soft key. The following functions are available:

① Dispaly part program directory: (page 396) — Displays the directory of part program numbers that are stored in NC memory.

② Call part program (page 404): — Calls a part program to edit in the editing job.

③ Erase part program: (page 406) — Erases a part program.

④ Rename part program: (page 407) — Alters a part program number.

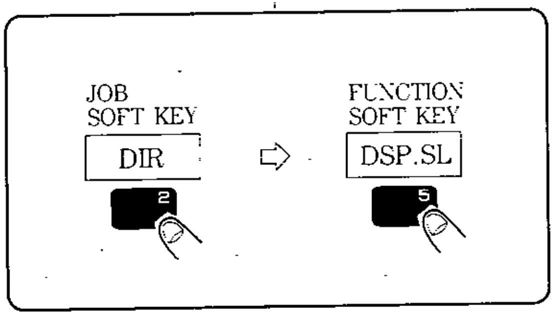
⑤ Copy part program: (page 409) — Copies a part program.

Details of the above functions are explained in the following:

(1) Display part program directory

This function displays the directory of part program numbers that are stored in NC memory. Up to 10 program numbers with comment and date can be displayed on the full screen. Up to 50 program numbers without comment or date can be displayed on the full screen. To turn on or off display of comment and date, use the [DSP.SL] function soft key.

KEYPOINT



1. Depress the [DIR] job soft key.

   The directory of part program numbers is displayed.

2. Depress the [DSP.SL] function soft key.

   If comment and date are displayed, they disappear when the key is depressed. If comment and date are not displayed, they appear when the key is depressed. If the directory of subprograms are displayed, comment and date cannot be displayed.

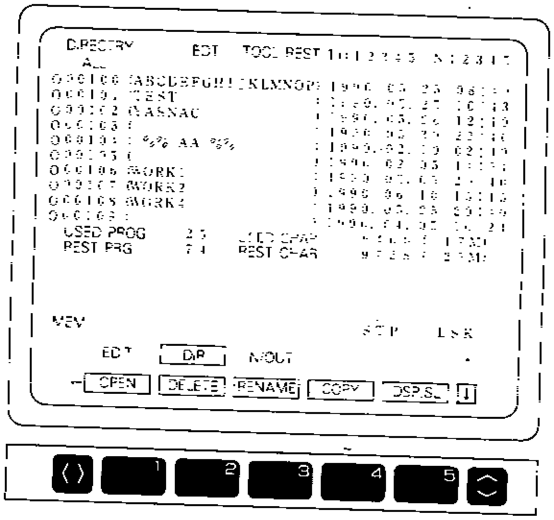


Fig. 3.4.23 Display of Part Program Directory with Comment and Date

Depressing the [DSP.SL] function soft key changes the screen as follows:

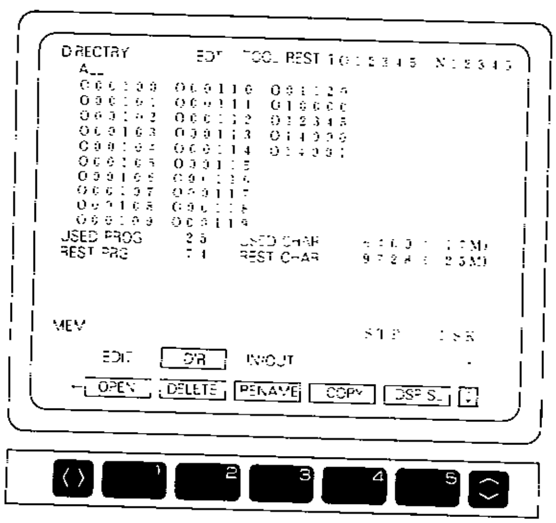
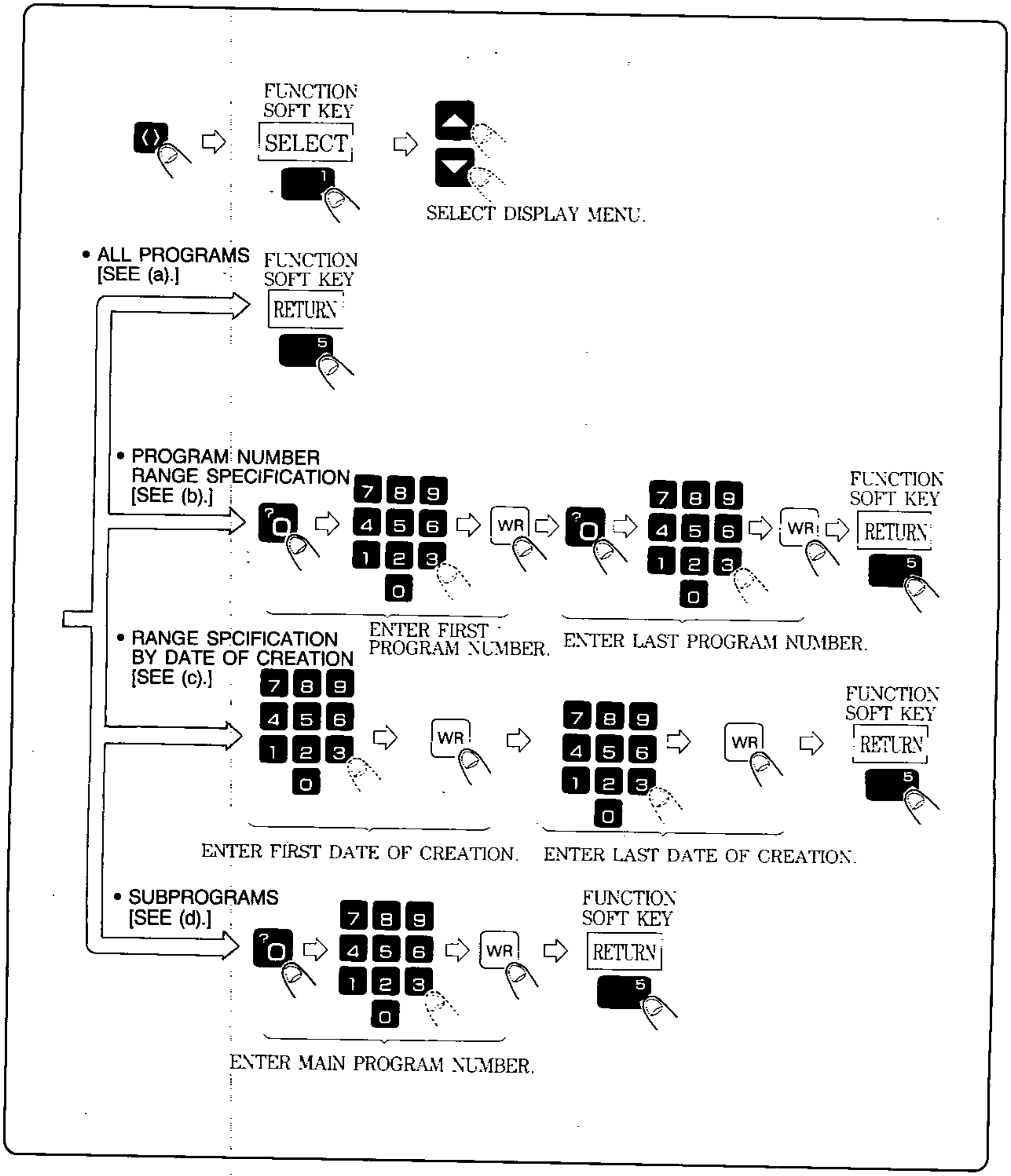


Fig. 3.4.24 Display of Part Program Directory without Comment or Date

There are the following four display method menus. Select one by the [SELECT] function soft key.

• All programs:	Display part program numbers that are stored.
• Program number range: specification	Display part program numbers in the specified range
• Range specification by: date of creation	Display part program numbers in the specified range of date of creation.
• Subprograms:	Display all part programs that are called by the specified main program.

KEYPOINT



FUNCTION SOFT KEY
SELECT
SELECT DISPLAY MENU.
• ALL PROGRAMS [SEE (a).]
FUNCTION SOFT KEY
RETURN
• PROGRAM NUMBER RANGE SPECIFICATION [SEE (b).]
WR
FUNCTION SOFT KEY
RETURN
ENTER FIRST PROGRAM NUMBER.
ENTER LAST PROGRAM NUMBER.
• RANGE SPCIFICATION BY DATE OF CREATION [SEE (c).]
WR
FUNCTION SOFT KEY
RETURN
ENTER FIRST DATE OF CREATION.
ENTER LAST DATE OF CREATION.
• SUBPROGRAMS [SEE (d).]
WR
FUNCTION SOFT KEY
RETURN
ENTER MAIN PROGRAM NUMBER.

3

(a) All programs

1. Set display selection to "✻ALL".
   In the initial state, "✻*ALL" is selected.

2. Depress the [RETURN] function soft key.

The screen returns to the directory screen and displays the directory of all program numbers.

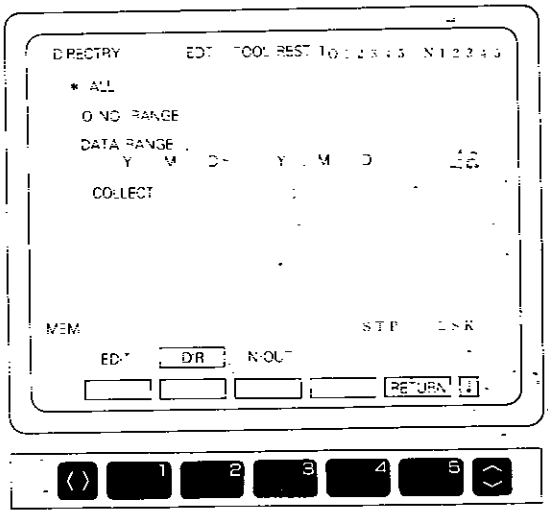


Fig. 3.4.25 Selecting "All Programs"

(b) Program number range specification

1. Set display selection to "O NO. RANGE".
2. Enter the first program number.

   (For example, O 1 0 0 0 WR)

3. Enter the last program number.

   (For example, O 2 0 0 0 WR)

4. Depress the [RETURN] function soft key.

   5

The screen returns to the directory screen and displays the directory of program numbers between the first and last program numbers.

NOTE If the input range of program numbers is invalid, a warning is displayed.

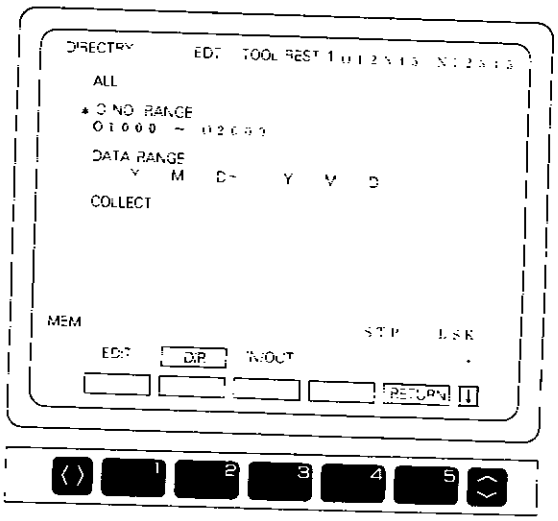


Fig. 3.4.26 Specification of Range of Program Numbers

(c) Range specification by date of creation

1. Set display selection to "DATE RANGE."
2. Enter the first date creation.

   (For example, )

   Entering the year only sets the first day of the year. Entering the year and the month only sets the first day of the month of the year.

3. Enter the last date of creation.

   (For example, )

   Entering the year only sets the last day of the year. Entering the year and the month only sets the last day of the month of the year.

4. Depress the [RETURN] function soft key.

   The screen returns to the directory screen and displays the directory of program numbers created in the specified period.

NOTE If the input range of date of creation is invalid, a warning is displayed.

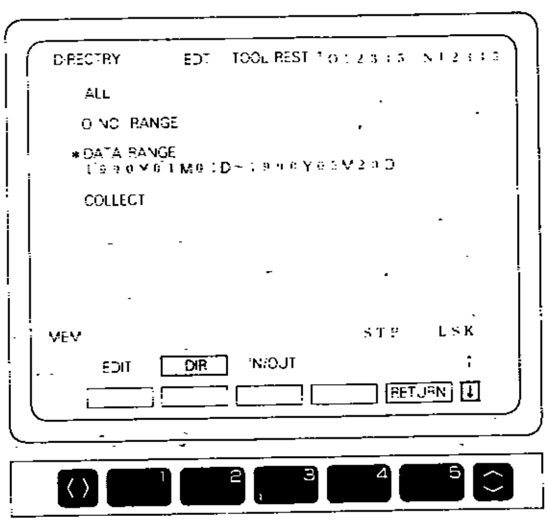


Fig. 3.4.27 Specification of Range of Date of Creation

(d) Subprograms

1 Set display selection to "COLLECT."

2 Enter the main program number.

(For example, )

3 Depress the [RETURN] function soft key.

The screen returns to the directory screen and displays the directory of programs that are called from the main program.

NOTES

1. When subprograms are displayed, only O-numbers can be displayed. Comment and date cannot be displayed and the [DSP.SL] key is disabled.

2. Programs that are called as subprograms but not stored are indicated with the program numbers blinking.

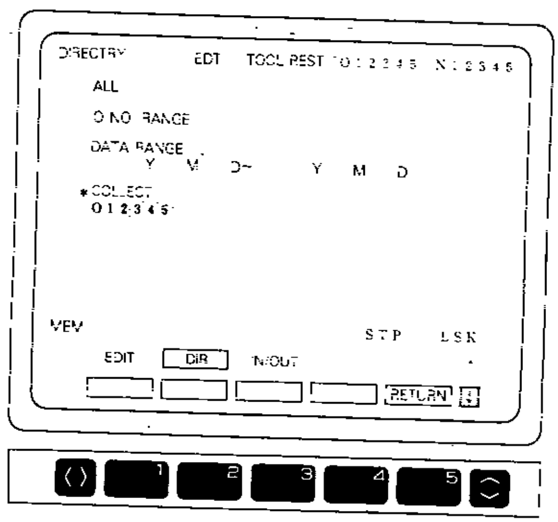


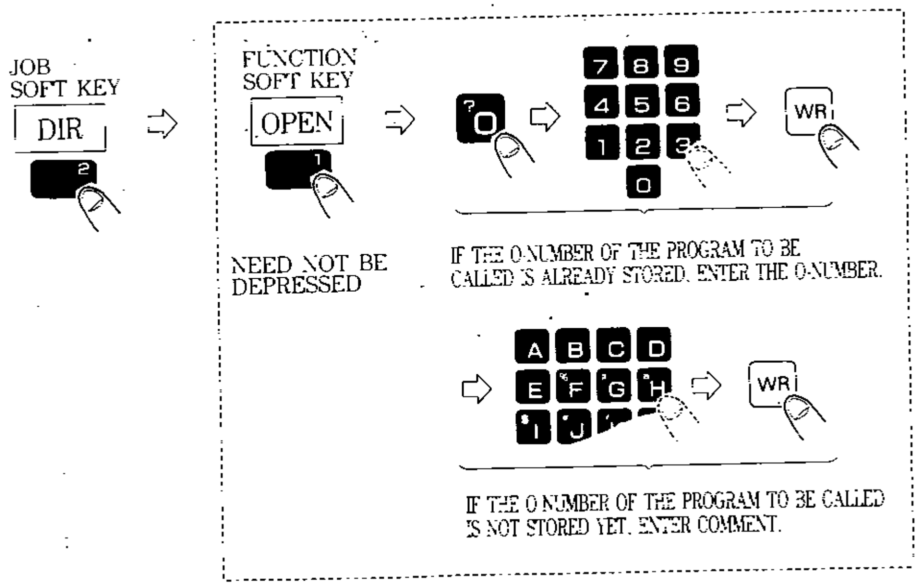
Fig. 3.4.28 Subprograms

(2) Calling part program

Part programs can be called from the program directory to the editing screen by simple operation. If the input program number has already been stored, the screen automatically changes to the editing screen and the program is ready to edit. If the input program number is not stored yet, enter comment. A new part program is created, the screen changes to the editing screen, then the program is ready to edit.

To call a part program, do the same as in the editing job. Programs can be called by depressing the [OPEN] function soft key, or by directly inputting the O-number and depressing the WRI key.

KEYPOINT



⇨ Up to three part program can be called by repeating the operation in the dotted line by an option.

The [OPEN] function soft key need not be depressed.

NOTES
1. New program numbers cannot be created in the editing prohibited state. Clear the editing prohibited state before calling programs.
2. Even running program can also be called, but cannot be edited.

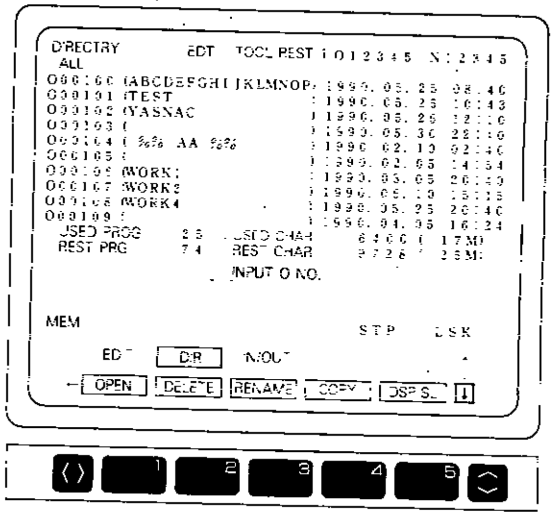


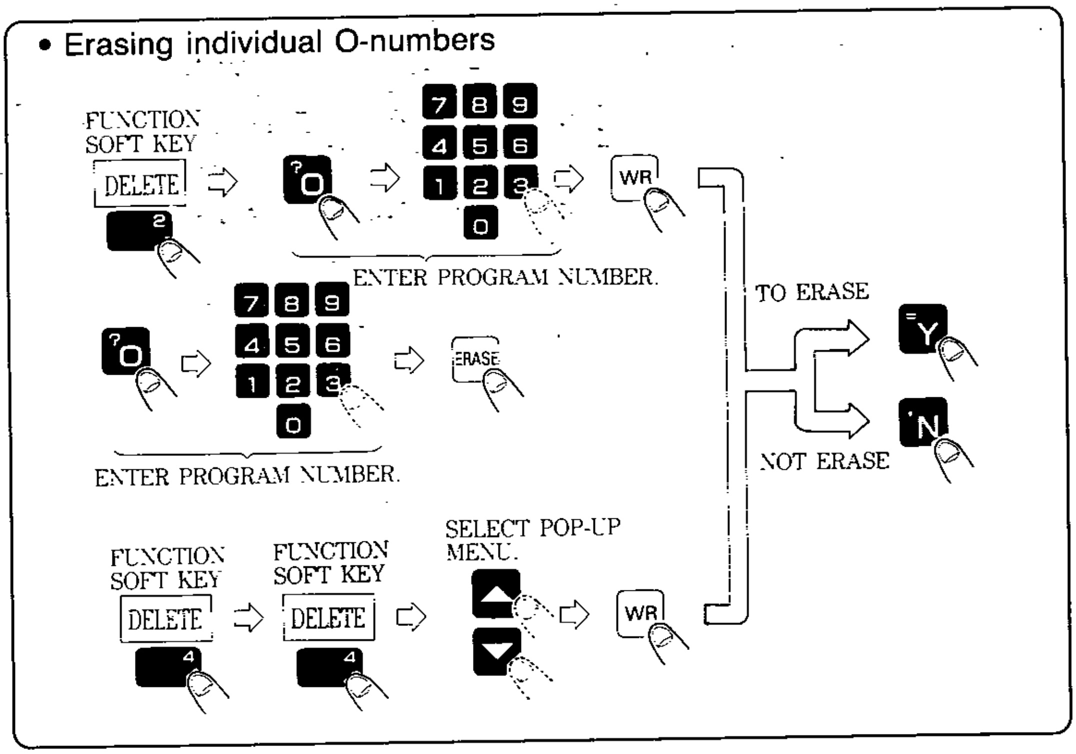
Fig. 3.4.29 Calling Part Program

(3) Erasing part progarms

Part programs can be erased from the program directory by simple operation. The operation is similar to erasing programs by editing job part program erase.

To erase a part program, do the same as in the editing job. Programs can be called by depressing the [DELETE] function soft key, or by directly input the O-number and depressing the ERASE key.

KEYPOINT



NOTES
1. Part programs cannot be erased in the editing prohibited state. Clear the editing prohibited state before erasing programs.
2. Running programs cannot be erased. Wait until the programs terminate.

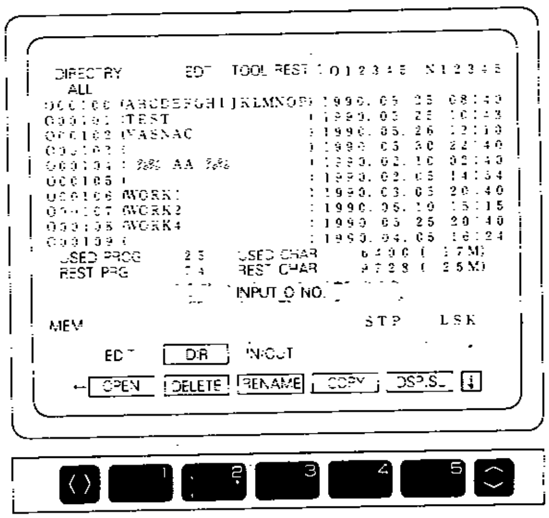
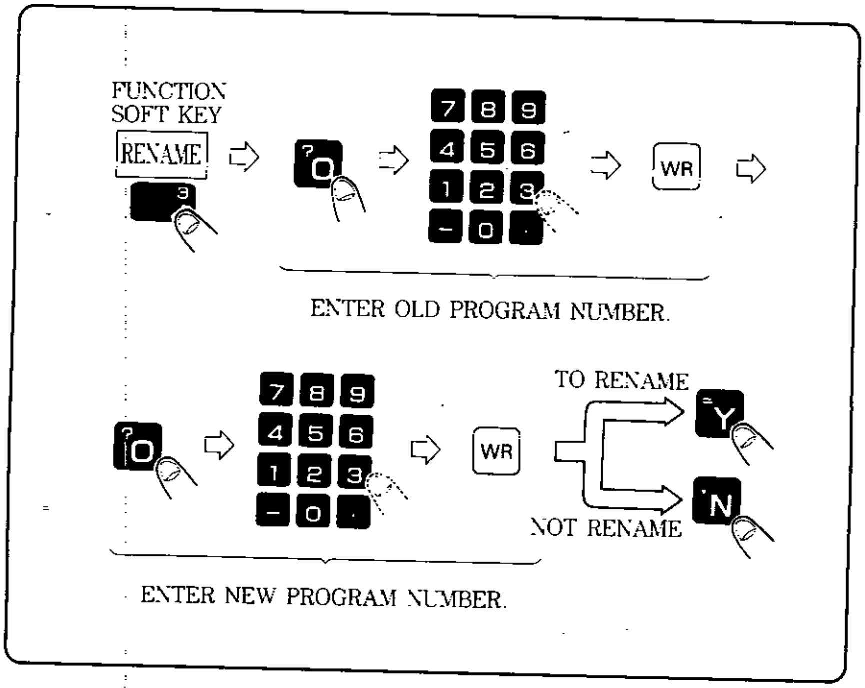


Fig. 3.4.30 Erasing Part Program

(4) Renaming part programs

Part program numbers can be changed by the following operation:

KEYPOINT



1. Depress the [RENAME] function soft key.

"INPUT OLD O NO." is displayed.

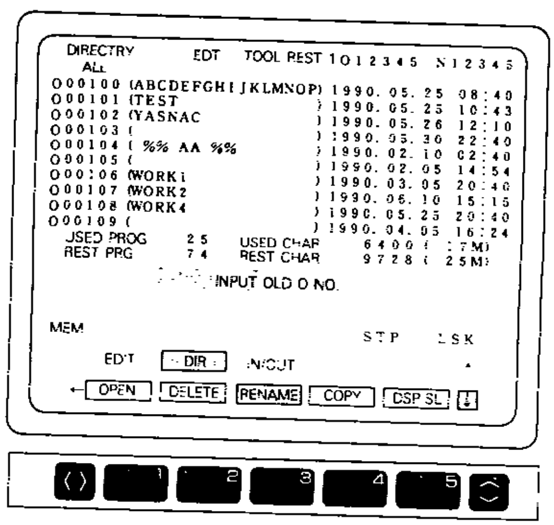


Fig. 3.4.31 Renaming Part Program

2 Enter an old program number.

(For example, O 1 0 0 WR)

"INPUT NEW O NO." is displayed.

3 Enter a new program number.

(For example, O 2 0 0 WR)

4 "RENAME OK? (Y/N)" is displayed.

- To rename the program, depress Y .
- Not to rename the program, depress N .

For a program with the program number at the beginning, the following parameter can determine whether the program number will be automatically edited when the program is renamed or copied [See (5).]

pm3005 D4 0: Does not edit the O-number automatically when the program is renamed, copied, or stored.

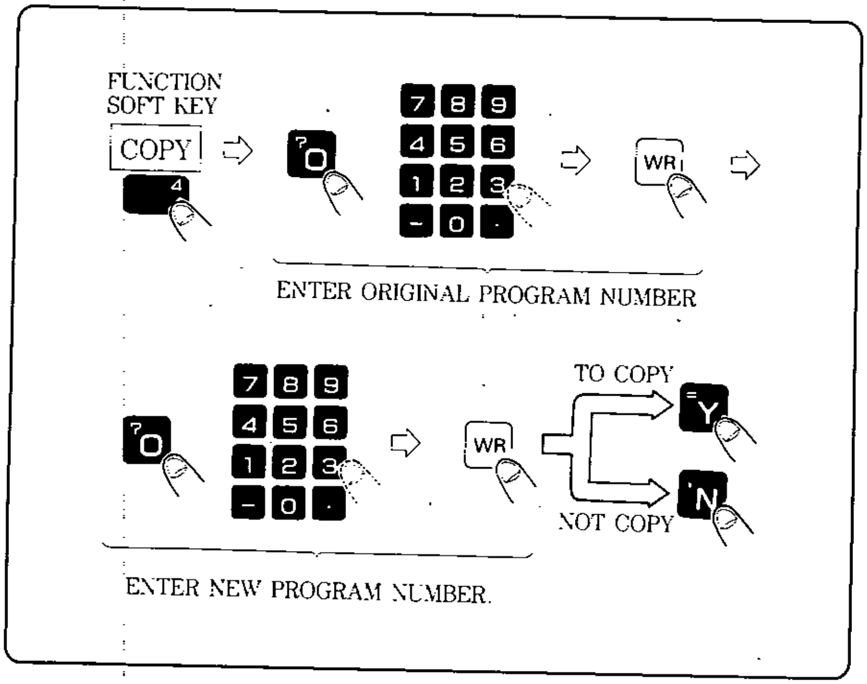
1: Edits the O-number automatically when the program is renamed, copied, or stored.

NOTE The date of the program is not updated when the program number is renamed.

(5) Copying part programs

Part programs can be copied by the following operation:

KEYPOINT



1 Depress the [COPY] function soft key.

"INPUT ORIG. O NO." is displayed.

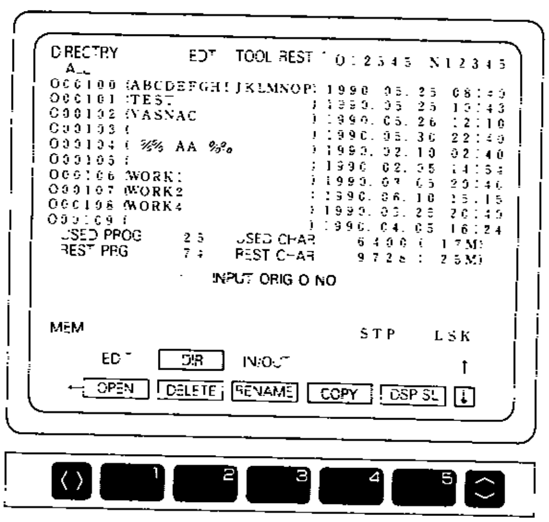


Fig. 3.4.32 Copying Part Program

2 Enter the original program number.

(For example, O 1 0 0 WR)

"INPUT NEW O NO." is displayed.

3 Enter the new program number to be created.

(For example, O 2 0 0 WR)

4 "COPY-OK? (Y/N)" is displayed.

- To copy the program, depress Y.
- Not to copy the program, depress N.

NOTE The new program created by copying has the new date of creation.

3.4.5.3 Part program I/O verification ([IN/OUT] job soft key)

Depress the [IN/OUT] job soft key. The following functions are available for the job.

① Part program input function: (page 412)	Stores part programs from external equipment to controller memory.
② Part program output function: (page 418)	Outputs part programs from controller memory to external equipment.
③ Part program verification: function (page 428)	Collates part programs in external equipment to that in controller memory.
④ BG reset function (page 433)	Input, output and verification can be aborted by the [BG.RESET] key.
⑤ I/O equipment setup function: (page 434)	Determines data I/O interface baud rate, stop bit length, and use of the control code.

Details of the above functions are described in the following.

NOTE In the editing prohibited status, part programs cannot be input, output, or verified. Clear the editing prohibited status in advance.

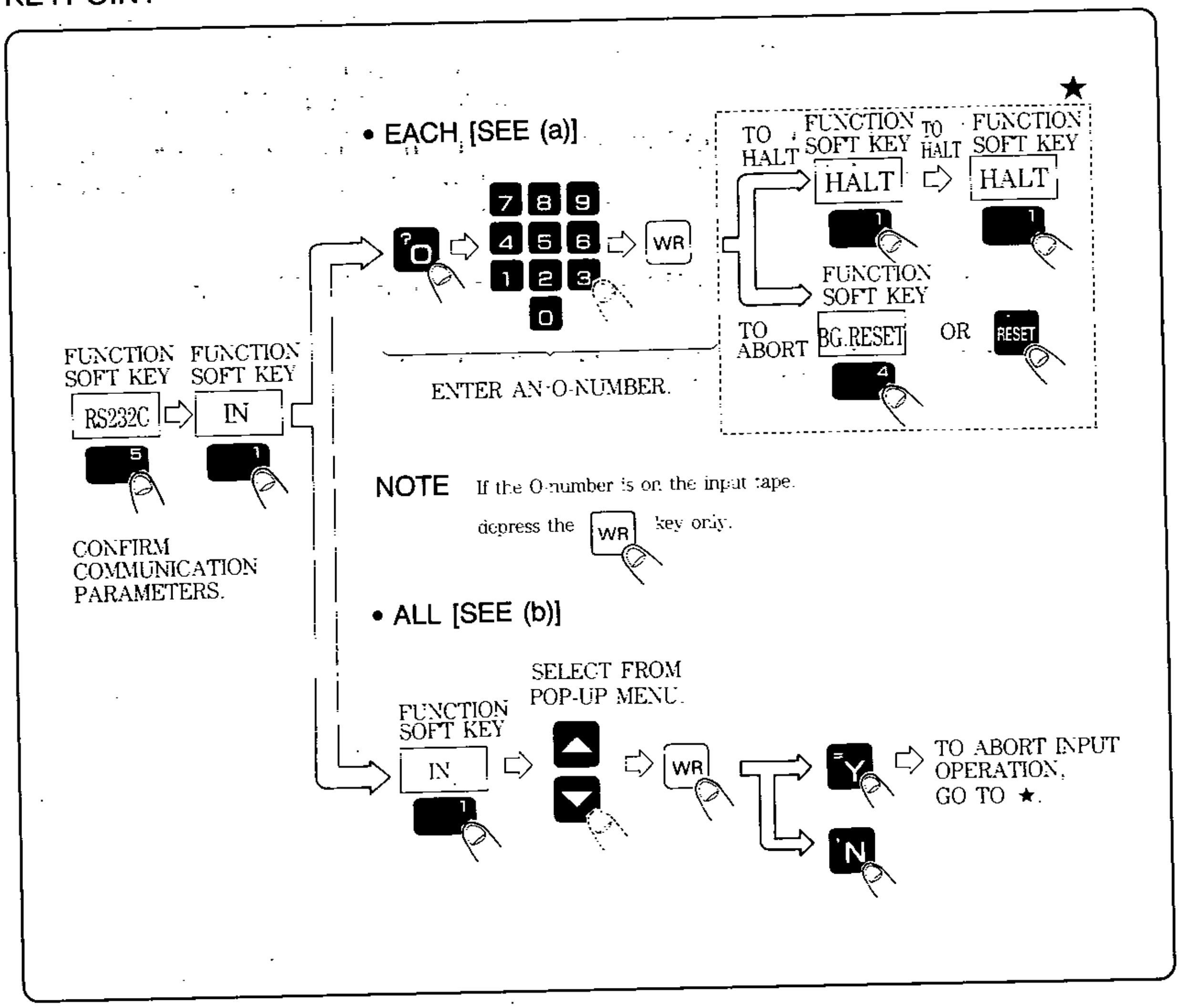
3

(1) **Part program input function**

There are EACH and ALL input menus, which can be selected from the pop-up menu.

- EACH: Enter part programs one by one.
- ALL: Enter all O-numbers on a single tape.

KEYPOINT



Depress the [RS232C] function soft key to confirm communication parameters.

(a) EACH

1 Depress the [IN] function soft key.

"INPUT (EACH)" is displayed.

2 Enter the O-number of the part program to be input.

(For example, 

)

- If the O-number is on the input tape, the number need not be entered from the screen. Depress the 

key only.

- Even if the O-number of the program is on the input tape, the program can be entered with a new O-number. Whether to change the O-number at the beginning of the part program can be determined by the parameter:

  pm3005 D4 0: Do not edit O-number automatically at program rename/copy/save.
  1: Edit O-number automatically at program rename/copy/save.

- If an input O-number has already been registered, a warning can be issued or the existing O-number can be deleted. Select either by the parameter:

  pm3005 D0 0: If the O-number input from tape is already registered, issue a warning.
  1: If the O-number input from tape is already registered, overwrite the existing program by the input program.

3 Program input is started and the part program number input screen shown in **Fig. 3.4.33** is displayed:

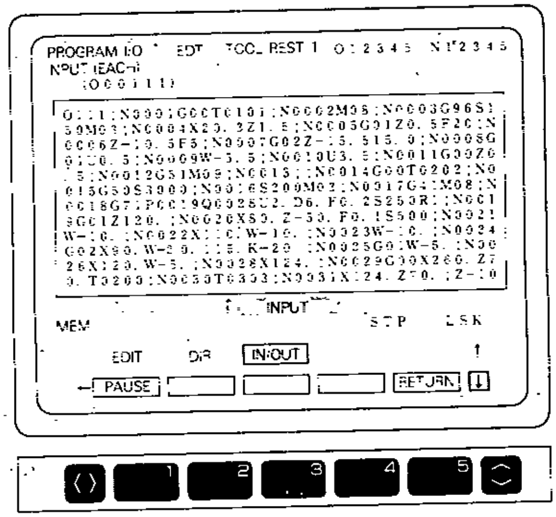


Fig. 3.4.33 Part Program Number Input Screen

4 To halt input operation, depress the [PAUSE] function soft key.

While input operation is halted, the [PAUSE] function soft key is highlighted.

To restart operation, depress the [PAUSE] function soft key again.

5 To abort operation, depress the [BG.RESET] function soft key or the RESET key.

Difference of the [BG.RESET] function soft key and the RESET key

- [BG.RESET] function soft key: Aborts only I/O or verification with tape. Operation of the program is not affected.
- RESET key: Resets all operations of the controller.

6 When a completion M code (M02, M30, or M99) or code % is found on the input tape, input operation is completed. After completion, "INPUT COMPLETE" is displayed.

NOTE The [RETURN] function soft key is disabled until input operation is completed. The process  keys are effective throughout the operation.

Whether a completion M code be regarded as the end of the program can be specified by the following parameter:

pm3005 D3 0: Do not assume the completion M codes (M02, M30, and M99) as the end of the program.

1: Assume the completion M codes (M02, M30, and M99) as the end of the program.

(b) ALL

1 Depress the [IN] function soft key.

Depress the [IN] function soft key again. "INPUT (EACH)" is displayed.

2 Depress the [IN] function soft key again.

The pop-up menu appears. (See **Fig. 3.4.34**.)

3 Depress cursor key to select ALL.

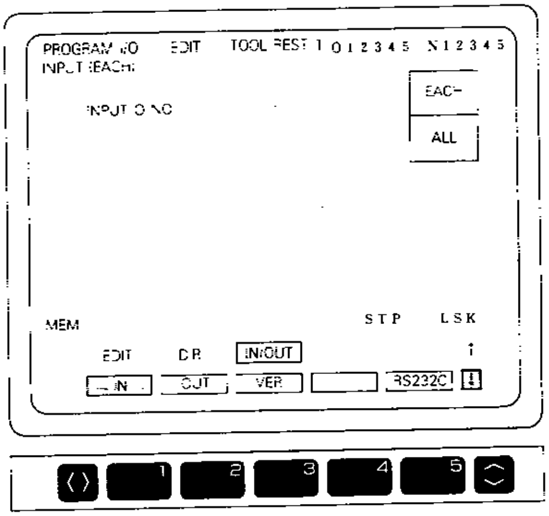


Fig. 3.4.34 Selecting ALL O-number Input

3 Depress the WR key.

The pop-up menu disappears and the ALL O-number input screen is displayed.

5 "ALL PROG. INPUT" and "INPUT OK? (Y/N)" are displayed as shown in **Fig. 3.4.35:**

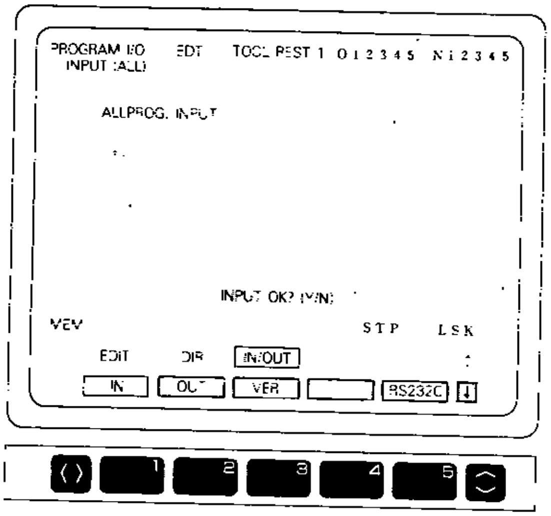


Fig. 3.4.35 ALL Input Screen

6 • To input programs, depress  .

Program input is started, the screen changes, and the contents of the part programs being input are displayed as shown in Fig. 3.4.33.

• To cancel input operation, depress  .

7 To halt or abort input operation, do the same as in (a), EACH O-number input.

8 The ALL O-number input operation is completed when code % is input. When the operation is completed, "INPUT COMPLETE" is displayed.

(2) Part program output function

The following six output menus can be selected from the pop-up menu:

① EACH .... Part programs are output one by one.

② PLR. (plural). .... Specified two or more part programs are output.

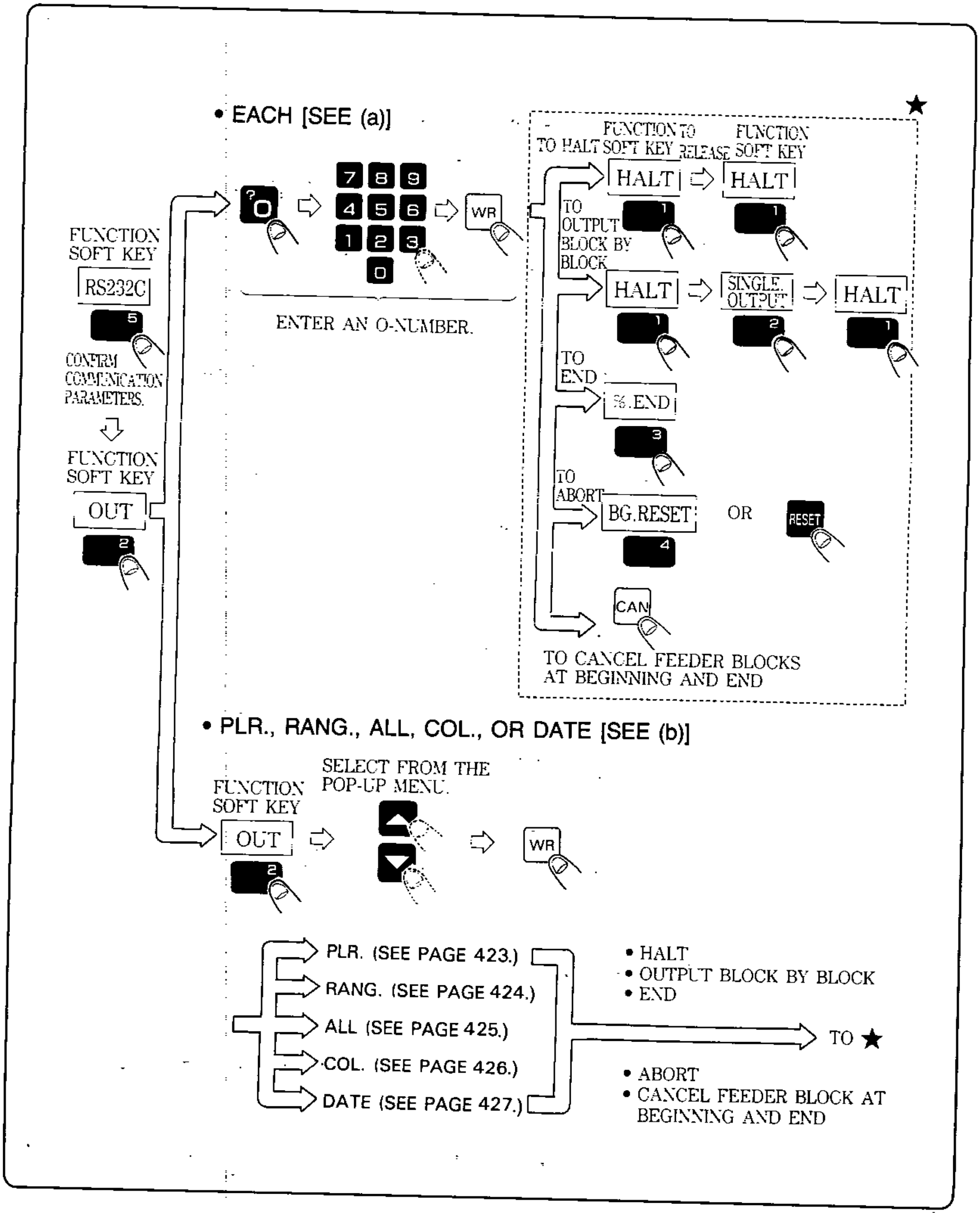
③ RANG (range)........ Part programs in the range between specified O-numbers are output.

④ ALL .... ALL part programs in controller memory are output.

⑤ COL. (correlated) ... Up the 500 subprograms that are called from a specified part program are output.

⑥ DATE .... Part programs created on a specified date are output.

KEYPOINT



• EACH [SEE (a)]
FUNCTION SOFT KEY
RS232C
CONFIRM COMMUNICATION PARAMETERS.
FUNCTION SOFT KEY
OUT
WR
ENTER AN O-NUMBER.
FUNCTION TO HALT SOFT KEY
FUNCTION TO RELEASE SOFT KEY
HALT
HALT
TO OUTPUT BLOCK BY BLOCK
HALT
SINGLE OUTPUT
HALT
TO END
%.END
TO ABORT
BG.RESET
OR
RESET
CAN
TO CANCEL FEEDER BLOCKS AT BEGINNING AND END
• PLR., RANG., ALL, COL., OR DATE [SEE (b)]
FUNCTION SOFT KEY
OUT
SELECT FROM THE POP-UP MENU.
WR
PLR. (SEE PAGE 423.)
RANG. (SEE PAGE 424.)
ALL (SEE PAGE 425.)
COL. (SEE PAGE 426.)
DATE (SEE PAGE 427.)
• HALT
• OUTPUT BLOCK BY BLOCK
• END
TO ★
• ABORT
• CANCEL FEEDER BLOCK AT BEGINNING AND END

Depress the [RS232C] function soft key to confirm communication parameters.

(a) EACH

1 Depress the [OUT] function soft key.
"OUTPUT (EACH)" is displayed.

2 Enter the O-number of the progrm to be output.
(For example, )

3 Program output is started, and the screen changes and displays the contents of the part program being output, as shown in **Fig. 3.4.36**.

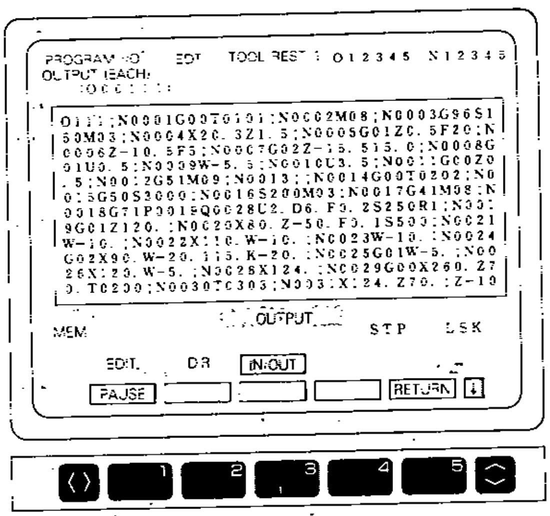


Fig. 3.4.36 O-number Output Screen

4 ⟨Halt⟩

To halt output operation, depress the [PAUSE] function soft key.


While output operation is halted, the [PAUSE] function soft key is highlighted.

To restart operation, depress the [PAUSE] function soft key again. 

⟨Output NC statements block by block⟩

To output NC statements block by block, depress the [PAUSE] function soft key, then depress the [SINGLE.OUTPUT] function soft key. Each time the [SINGLE.OUTPUT] function soft key is depressed, one block up to an EOB (;) is output. To restart operation, depress the [HALT] function soft key again.

⟨End⟩

To end output operation, depress the [%.END] function soft key.

"%" and the feeder block are added after the output data so that the output can be used for NC format tape.

⟨Abort⟩

To abort output operation, depress the [BG.RESET] function soft key or the RESET key.

⟨Cancel feeder blocks at beginning and end of output data⟩

To cancel the feeder blocks at the beginning and end of output data, depress the CAN key during feeder output.

5 When output operation is completed, "OUTPUT COMPLETE" is displayed.

NOTE The [RETURN] function soft key is disabled until output operation is completed.

The process keys are effective throughout the operation.

(b) PLR., RANG., ALL, COL., and DATE

1 Depress the [OUT] function soft key.

"OUTPUT (EACH)" is displayed.

2 Depress the the [OUT] function soft key again.

The pop-up menu appears to provide choice from EACH, PLR., RANG, ALL, COL., and DATE. (See Fig. 3.4.37.)

3 Depress cursor key ▲ or ▼ to select an output pop-up menu.

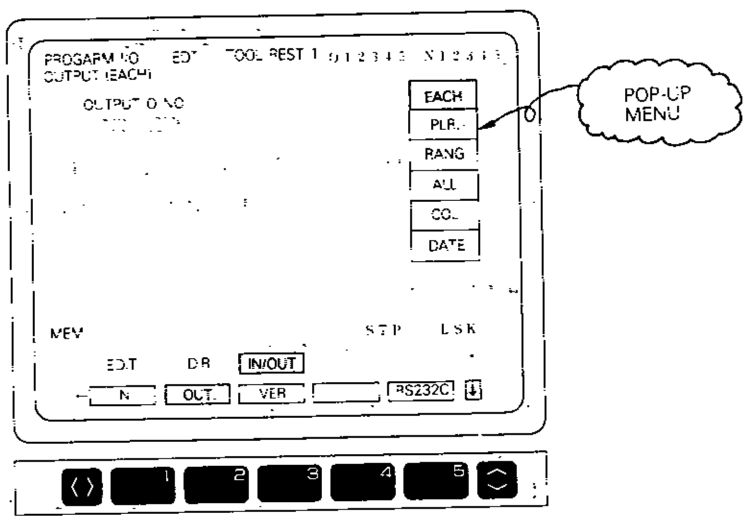


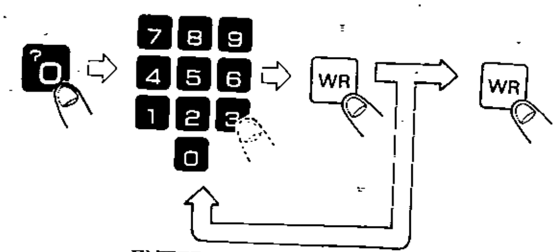
Fig. 3.4.37 Output Menu Select Screen

4 Depress the WR key.

The output pop-up menu disappears and the specified output menu is displayed.

(i) PLR.

KEYPOINT



ENTER UP TO 20 O-NUMBERS.

5 Enter the O-number of the program to be output.

(For example, )

The cursor moves to the next line. Up to 20 O-numbers can be entered consecutively.

6 Depress the  key.

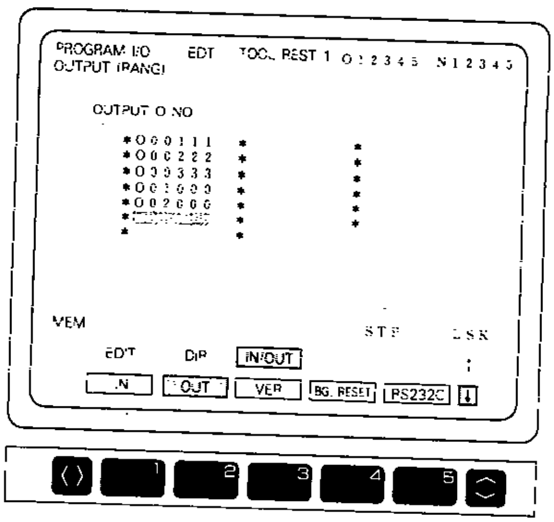
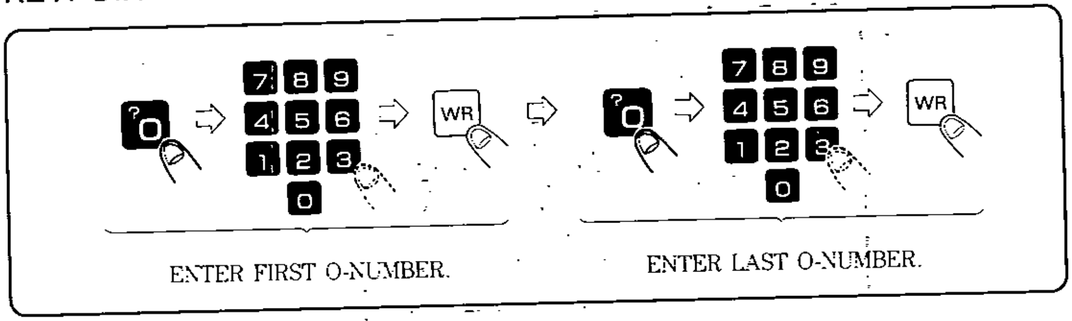


Fig. 3.4.38 O-number Input Screen

7 Program output is started and the contents of the part program being output are displayed. The specified programs are output on a single NC tape.

(ii) RANG.

KEYPOINT



5 The operator is requested to enter the first and last O-numbers of the range to be output.

For example

for the first O-number, and

for the last O-number

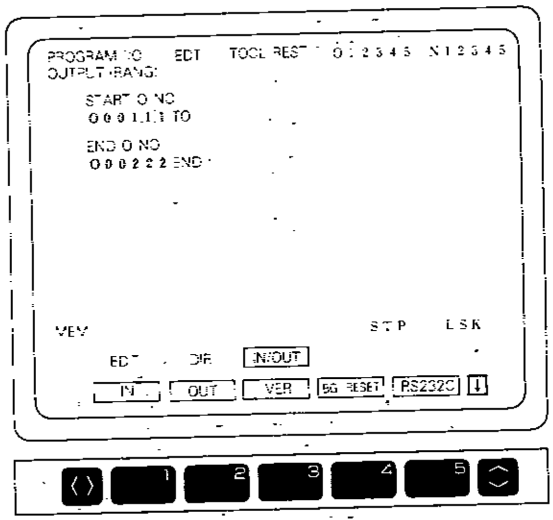


Fig. 3.4.39 Screen for Entering First and Last O-numbers

6 Program output is started and the contents of the part program being output are displayed. Part programs between the first and the last O-number are output onto a single NC tape.

If no O-number was found in the range, a warning is issued.

(iii) ALL

KEYPOINT

To output	Y
Not to output	N

5 "ALL PROG. OUTPUT" and "OUTPUT OK? (Y/N)" are displayed. To start output operation, depress Y .

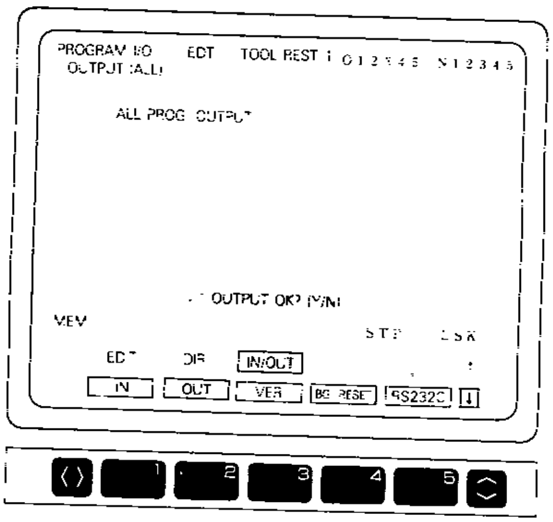


Fig. 3.4.40 ALL Output Screen

6 Program output is started and the contents or the part program being output are displayed.

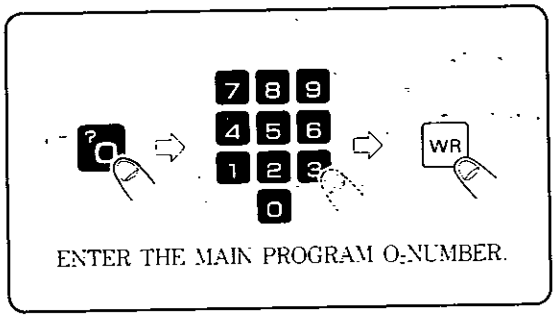
All the part programs that are registered are output onto a single NC tape.

7 Not to start output operation, depress N .

(iv) COL.

Specify the main program among part programs to output the subprograms, program copies, and macro programs that are called in the main program.

KEYPOINT



5 The operator is requested to enter an O-number. Enter the O-number of the main program to output.

(For example, )

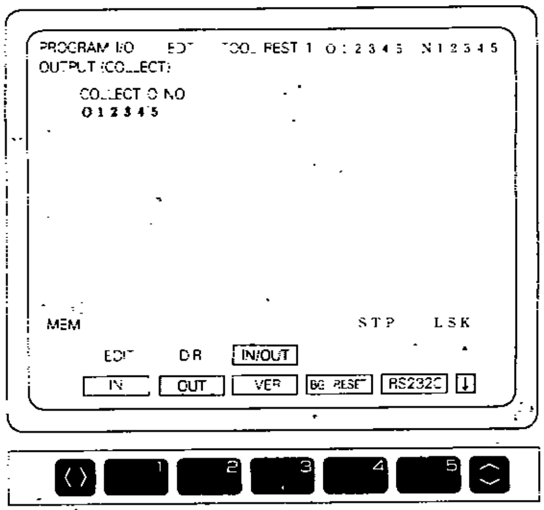


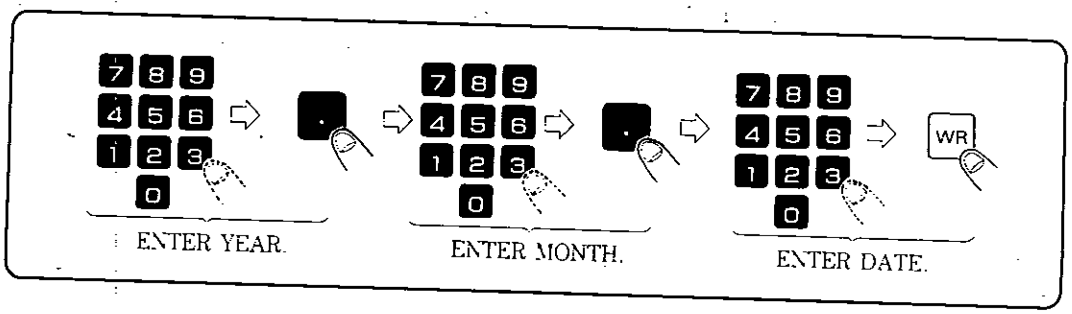
Fig. 3.4.41 O-number Input Screen

6 Program output is started and the contents of the part program being output are displayed. The main program and the function programs called by the main program output onto a single NC tape.

(v) DATE

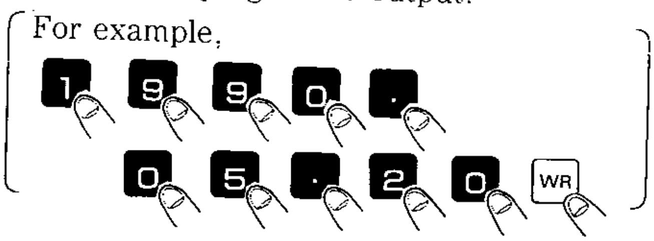
The date of the creation and registration of a part program is kept in controller internal memory. A part program can be output using this date.

KEYPOINT



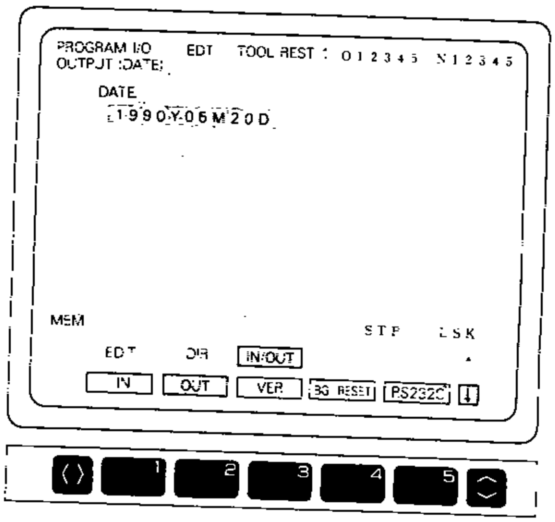
5 The operator is requested to enter the date. Enter the date of creation of the part program to output.

For example,



If only the year is entered, all the programs created in that year are output. If only the year and the month are entered, all the programs created in that month of that year are output.

Fig. 3.4.42 Date Input Screen



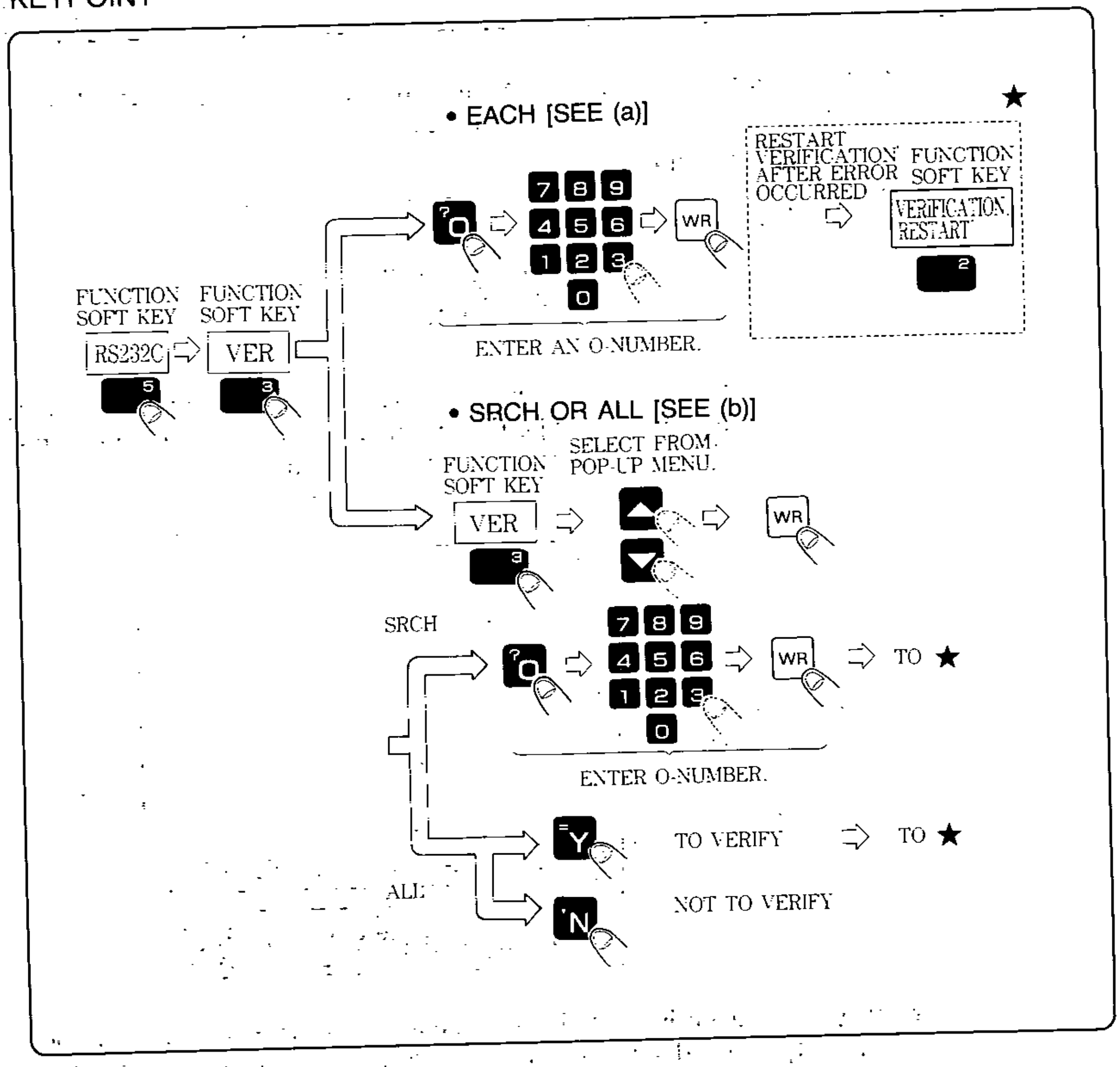
6 Program output is started and the contents of the part program being output are displayed. Part programs created on the specified date are output onto a single NC tape.

(3) Part program verification function

The following three verification menus can be selected from the pop-up menu:

- EACH .... Part programs are verified one by one.
- SRCH (search) .. If more than one O-number is on a single tape, a specified O-number is searched for automatically, and only that O-number is verified.
- ALL .... All O-numbers on a single tape are verified.

KEYPOINT



Deprees the [RS232C] function soft key to confirm communication parameters.

(a) EACH

1 Depress the [VER] function soft key.

2 Enter the O-number of the part program to verify.
(For example, O 1 1 1 WR)

3 Verification is started and the contents of the input tape part program to be verified.

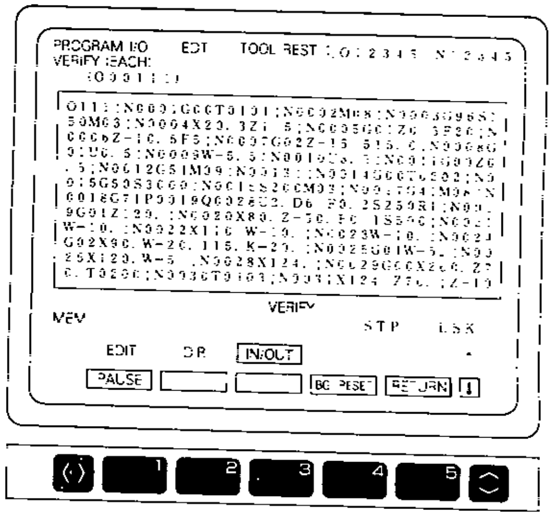


Fig. 3.4.43 O-number Verification Screen

4 If inconsistency between the memory contents and the input from external equipment was found, the screen is divided into two halves. The upper half displays the input from tape. The lower half displays controller internal data. The last EOBs (;) of the unmatched blocks blink in highlight.

A verification error also occurs if the input from tape is longer than memory contents.

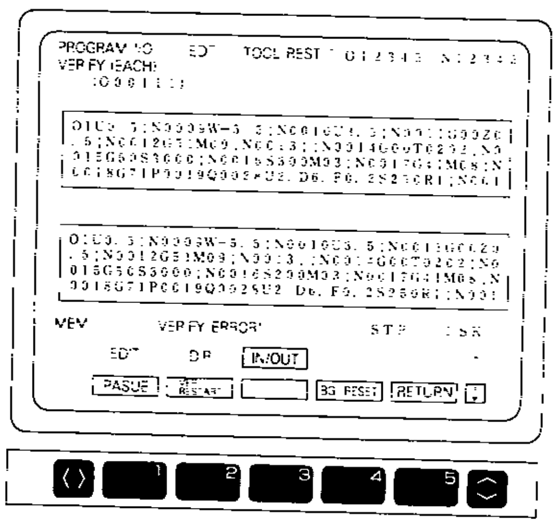


Fig. 3.4.44 Display of Mismatch

5 To restart verification, depress the [VER.RESTART] key.

Verification is restarted from the beginning of the block next to the one where the verification error occurred. If a verification error occurred in the last block, verification cannot be restarted.

(b) SRCH and ALL

1 Depress the [VER] function soft key.

2 Depress the [VER] function soft key again.

The pop-up menu appears and the operator is requested to select EACH, SRCH, or ALL. (See **Fig. 3.4.45**.)

3 Depress the cursor key  or  to select from the pop-up menu.

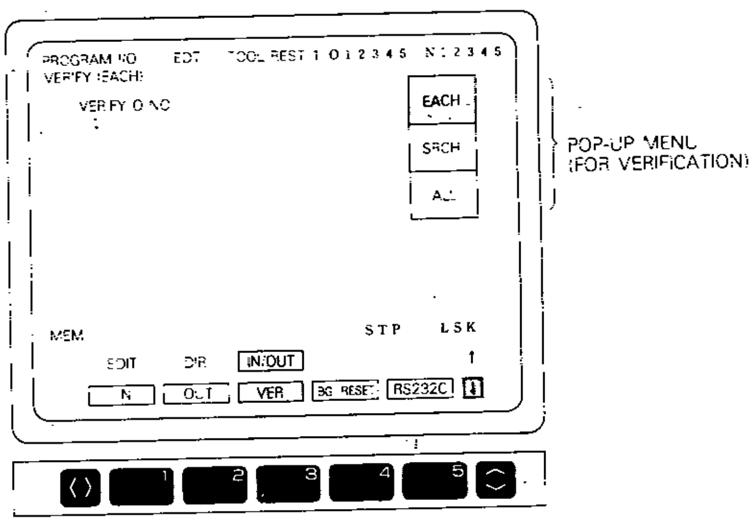


Fig. 3.4.45 Verification Menu Select Screen

4 Depress the WR key.

The pop-up menu disappears and the specified verification menu is displayed.

(i) SRCH

5 Enter the O-number of the part program to verify.

(For example, [O] [1] [1] [1] [WR])

6 The whole tape is searched for the input O-number. When the number is found, verification is started from that O-number, and the contents of the verified part program are displayed as shown in Fig. 3.4.43.

NOTE If an invalid O-number is input, "NOT FOUND O NO!" is displayed. Depress the [CAN] key and restart from **1**.

7 If inconsistency between the memory contents and the input from external equipment was found, the screen changes as shown in Fig. 3.4.44. The last EOBs (;) of the unmatched blocks blink in highlight.

(ii) ALL

5 "ALL PROG. VERIFY" and "VERIFY OK? (Y/N)" are displayed.

To start verification, depress Y.

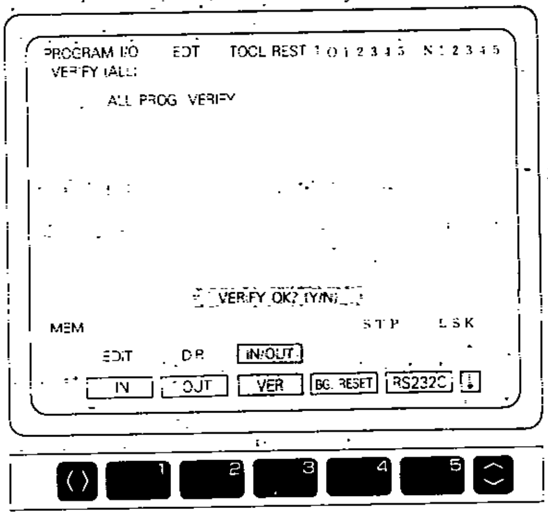


Fig. 3.4.46 ALL O-number Verification Screen

6 Program verification is started, the screen changes, and the contents of the part programs being verified are displayed.

7 To cancel verification, depress N.

8 If inconsistency between the memory contents and the input from external equipment was found, the screen shown in Fig. 3.4.44 is displayed, and the last EOBs (;) of the unmatched blocks blink in highlight.

(4) **BG reset function**

This is the background reset switch, which functions differently from the 

switch on the operator panel.

The BG reset switch performs the following:

- Aborts I/O verification operation
- Turns on the label skip function
- Releases background alarms (ALM9000 to 9999)

The [BG.RESET] function soft key 

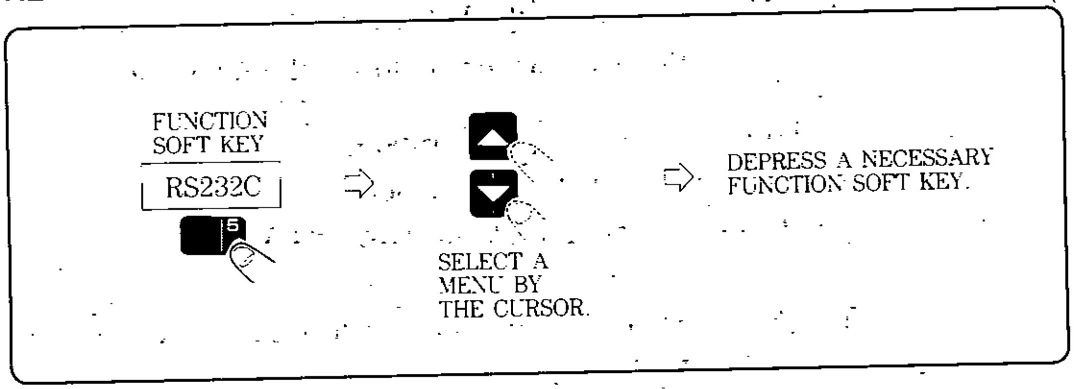
is disregarded if depressed during operation.

The reset status indication on the screen (RST) is not displayed even if the [BG.RESET] function soft key 

is depressed.

(5) I/O equipment setup function

KEYPOINT



1 Depress the [RS232C] function soft key.

The screen in **Fig. 3.4.47** appears. The screen is used to display and set up communication parameters for interfacing with external equipment according to RS-232C. There are the first and second ports for input and output.

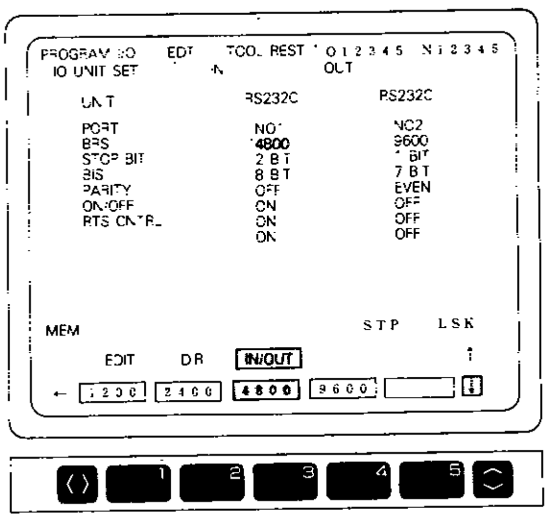


Fig. 3.4.47 I/O Equipment Setup Screen

2 Place the cursor at a menu item to set up by cursor key ▲ or ▼.

Different function soft keys appear as the selected menu item changes.

3 Select and depress a soft key.

Date are written to the cursor location.

(Example) In baud rate setup, depressing the 9600 soft key sets 9600 baud.

NOTE "****" indicates that the item is automatically set up and cannot be changed.

Table 3.4.2 lists possible setting for serial interface:

Table 3.4.2 Serial Interface Setting

Menu Item	Input	Output
Equipment	YE tape reader	
	General-purpose RS-232C	General-purpose RS-232C
Port	First	First
	Second	Second
Baud Rate	100 or 75	100 or 75
	110 or 150	110 or 150
	300	300
	600	600
	1200	1200
	2400	2400
	4800	4800
	9600	9600
Stop Bit	1 bit	1 bit
	2 bits	2 bits
Bit Length	7 bits	7 bits
	8 bits	8 bits
Parity Check Communication	Even parity	Even parity
	Odd parity	Odd parity
	No	No
Control Code	Yes	Yes
	No	No
RTS Control	Yes	Yes
	No	No
Parity Check ISO	Yes	Yes
	No	No

NOTE 1. The two RS-232 ports cannot be used at the same time. Settings for the first and second ports are independent from each other.

2. For communication by the baud rate 9600 bps, use following parameters.

Baud Rate	9600 bps
Stop Bit	2 bits
Bit Length	8 bits
Parity Check Communication	No/Even/Odd
No data inherent control	No.

The menu items are explained in the following:

(a) **Equipment**

Select YE tape reader or YE tape puncher when a YASNAC specialized tape reader or tape puncher is to be connected. If YE tape reader or YE tape puncher is selected, the baud rate and the following items are set up automatically, and "****" is displayed for these items.

Automatic setting: 4800 baud,
stop bit = 2 bits,
bit length = 8 bits,
parity check communication = No,
control code = Yes

If general-purpose RS-232C is selected, various equipment can be connected by setting parameters.

(b) **Ports**

Select the first or second port.

(c) **Baud rate**

Set up the communication speed. In Table 3.4.2, there are two numbers each in the lowest two baud rate classes. Whether 100 and 110 or 75 and 150 are available is determined by the parameters pm0011D7, pm0013D7, pm0016D7, and pm0018D7.

If 1 is set for pm0011D7; 100 and 110 appear as soft keys for the first port input. If 0 is set for the parameter, 75 and 150 are displayed. The same rates apply to pm0013D7 with the first port output, pm0016D7 with the second port input, and pm0018D7 with the second port output.

(d) **Stop bit**

Set the stop bit length.

(e) **Bit length**

Set the data bit length.

(f) **Parity check communication**

Specify whether to use parity check by hardware.

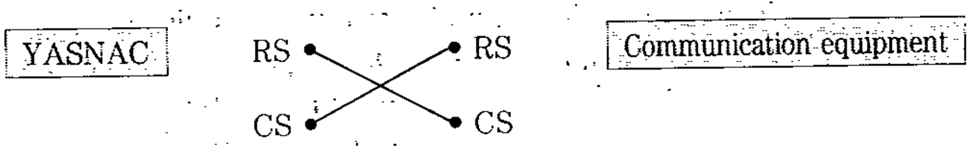
(g) **Control code**

Specify whether to use control code.

(h) RTS control

The RTS control governs YASNAC and connected communication equipment using RS and CS.

To performs RTS control, use RS-CS connection cables as shown in the figure:



(i) Parity check ISO

Specify whether to perform parity check with data higher bit at ISO code I/O. This item can be specified by the operator even if YE tape reader or YE tape puncher has been selected.

Other communication control parameters

- **DR line check**

	Input	Output
First port	pm0012 D2	pm0014 D2
Second port	pm0017 D2	pm0019 D2

When the corresponding parameter is 1, DR check is made for the specified input or output at specified port.

The DR line check tests whether the DR line of the remote equipment is on (that is, whether the equipment is ready). To perform the DR check, connect the remote equipment that supports the DR line control function to the NC controller and the remote equipment DR line with a cable.

(j) NC data independent control

Specify whether NC independent check or character codes are to be used. The following describes the diffrerence between when this control is provided and when it is not.

① At input

- When this control is provided
pm0004: D1, D2, D3 and
pm3005: D0, D2, D3, D4, D5 are as set by parameters.
pm4100 to 4109 and pm4144 to 4146 are validated.
EIA/ISO automatic judgement provided.

- When this control is not provided
  pm0004: × × × × 100 ×   pm3005: × × 0000 × 0
  The parts other than × are fixed.
  pm4100 to 4109 and pm4144 to 4146 are invalidated.
  Fixed to ISO.

② At output

- When this control is provided
  Other than pm0004: D3 and pm3005: D5 are set as by parameters.
  pm4100 to 4109 and pm4144 to 4146 are validated.
- When this control is not provided
  pm0004: 0110 × × × 0   pm3005: × × 0 × × × × ×
  The parts other than × are fixed.

3.4.6 [SET] Setup Process

The setup process is preparation for cutting. In this process, tool offset data and work coordinate system shift data can be displayed and set up. The data can be displayed or set up any time regardless of the controller operation mode.

1 Depress the  key. The [OFFSET] job soft key is displayed.

2 Depress the [OFFSET] job soft key. The offset job supports the tool

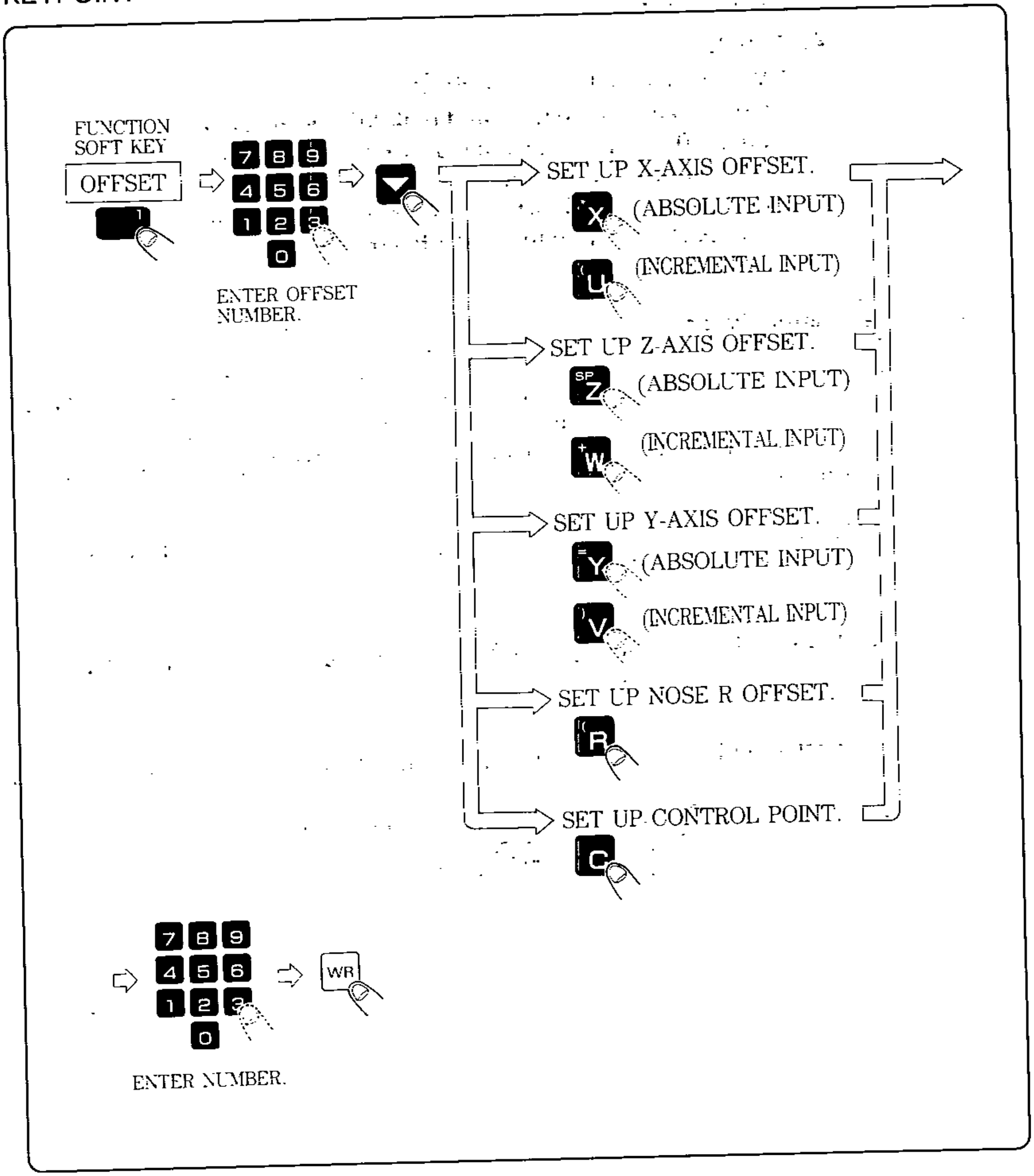
offset function and the work coordinate system shift function.

3 Select the [OFFSET] or [WORK] function soft key.
 

3.4.6.1 Offset function

This function sets up tool offset data.

KEYPOINT



There are the following offset data:

X	X-axis tool offset
Z	Z-axis tool offset
R.........	Nose R offset
C.........	Control point
Y.........	Y-axis tool offset (option)

The tool offset sets can be added by an option.

Tool offset	Standard:	16 sets
	Option 1:	99 sets
	Option 2:	299 sets

In a multi-program system, whether individual programs have separate tool offset areas can be determined by the parameter:

Parameter pm4020 D0
- 0: Offset numbers are not assigned to individual programs during multi-program operation.
- 1: Offset numbers are assigned to individual programs during multi-program operation.

When the parameter is set to 1 and each program has exclusive offset area, the total number of available sets are divided for the programs. Each program has the same tool numbers and offset numbers.

When the parameter is set to 0 and the programs share one offset area, the offset area is not divided. All programs share the unique tool numbers and offset numbers.

(1) Setting offset data

1. Use cursor key ▲ or ▼ to place the cursor at the offset number to be set.

   Otherwise, enter the offset number to search for.

   (For example, 0 1 ▼).

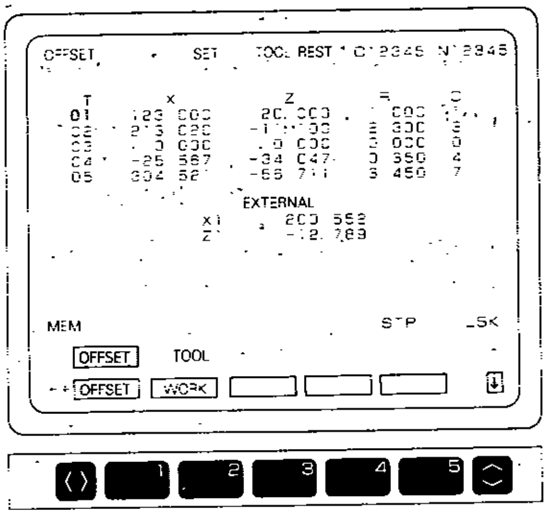


Fig. 3.4.48 Offset Screen (2-axis Specification)

2. The cursor moves to the searched offset number. Five sets of tool offset, nose R offset, and control point including the searched offset number are displayed.
3. Enter number.

• To set X-axis offset

X 1 2 . 3 4 5 WR .... Absolute input

U 1 2 . 3 4 5 WR .... Incremental input

• To set Z-axis offset

Z ...... Absolute input

W ..... Incremental input

- **To set Y-axis offset**

..... Absolute input

..... Incremental input

- **To set nose R offset**

- **To set control point**

NOTE R and C cannot be input with an incremental value.
The number of tool offset number sets can be increased by option.
For 3-axis specification, external coordinates are not displayed.

(2) Clearing offset data

1. Depress  key.

2. Depress the [0 CLR] function soft key  to set zero for the offset data at the offset number indicated by the cursor.

3. Depress the [ALL CLR] function soft key . "CLEAR ALL OFFSET? (Y/N)" is displayed.

Depress  to set zero for all offset memory.

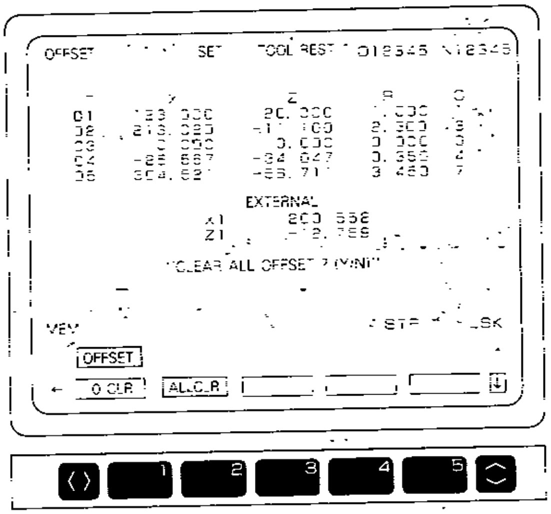


Fig. 3.4.49 Offset Screen ([ALL CLR] has been Depressed)

3.4.6.2 Work coordinate system shift function

Work coordinate system shift data can be set up.

The work coordinate system shift data are stored to offset number T00. This value takes effect on every tool when the work coordinate system is set up.

In a multi-program system, the work coordinate system shift data is determined for each program, regardless of the parameter that determines whether the tool offset area is shared by the programs.

(1) Setting work coordinate system shift data

Enter number after the address to set up.

- **To set X-axis shift data**

.... Absolute input

.... Incremental input

- **To set Z-axis shift data**

...... Absolute input

..... Incremental input

- **To set Y-axis shift data**

..... Absolute input

..... Incremental input

- **To set C-axis shift data**

...... Absolute input

..... Incremental input

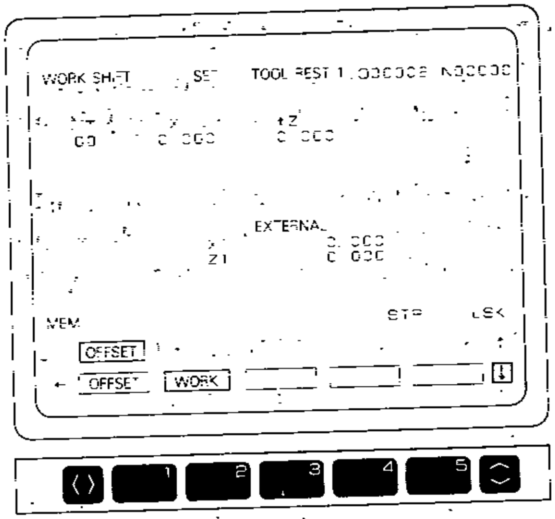


Fig. 3.4.50 Work Shift Screen (2-axis Specification)

(2) Clearing work shift data

1. Depress key 

2. Depress the [0 CLR] function soft key to set zero for the work shift data. 

3.4.7 Run Process [RUN]

Created part programs are executed in the run process. During the process, status of program operation and command values of the running program are displayed. Also, internal switches can be displayed and set up.

The above jobs are classified as the program job, command value job, and setting job.

1 Depress the [RUN] key. Any of the following jobs is displayed:

- Program job ———— See Par. 3.4.7.1 (page 448.)
- Command value job—— See Par. 3.4.7.2 (page 463.)
- Setup job ———— See Par. 3.4.7.3 (page 464.)

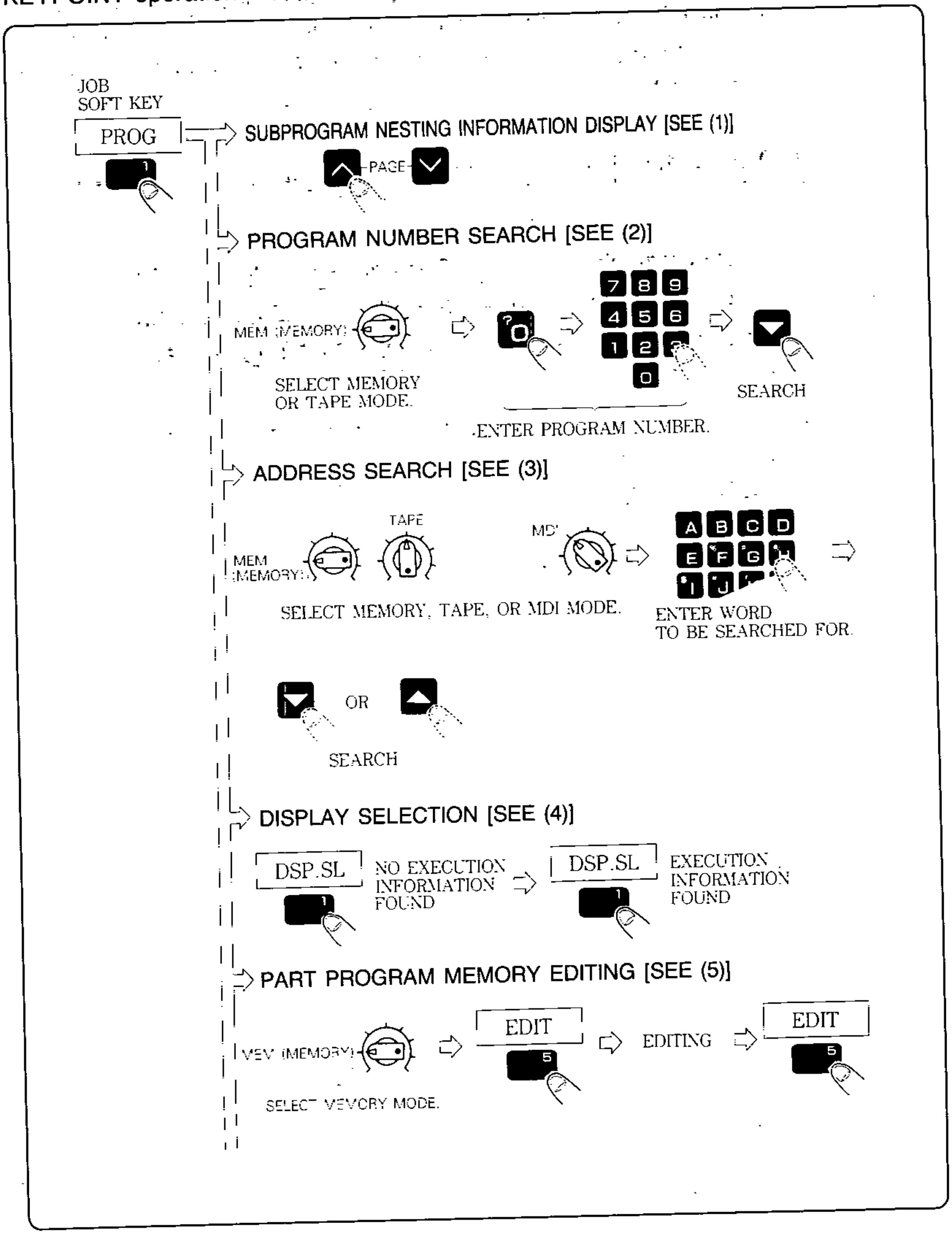
2 To display a necessary job, depress the corresponding job soft key [PROG], [COMMND], [SETING] or [RUN].

1 2 3

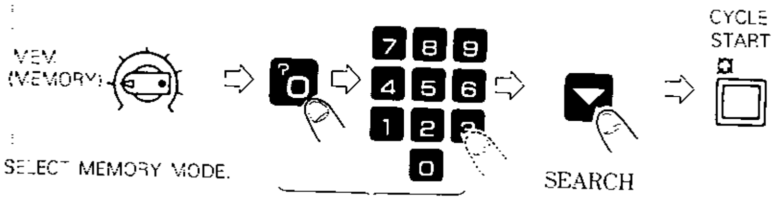
Otherwise, depress the key to switch the displayed job.

3.4.7.1 Program job

KEYPOINT operation



PART PROGRAM OPERATION[SEE (6)]



SELECT MEMORY MODE.

SEARCH

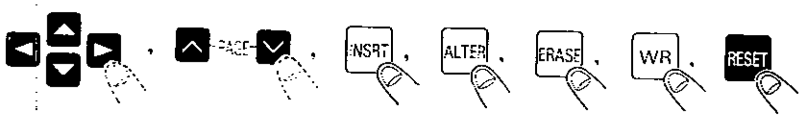
ENTER PROGRAM NUMBER.

MDI PROGRAM INPUT [SEE (7)]

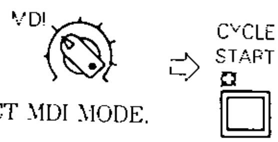
SELECT MDI MODE.

ENTER MDI PROGRAM.

MDI PROGRAM EDITING [SEE (8)]



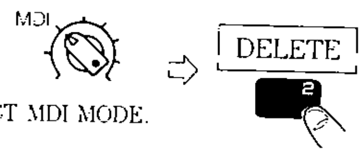
MDI PROGRAM OPERATION [SEE (9)]



SELECT MDI MODE.

Place the cursor at the beginning of the MDI program

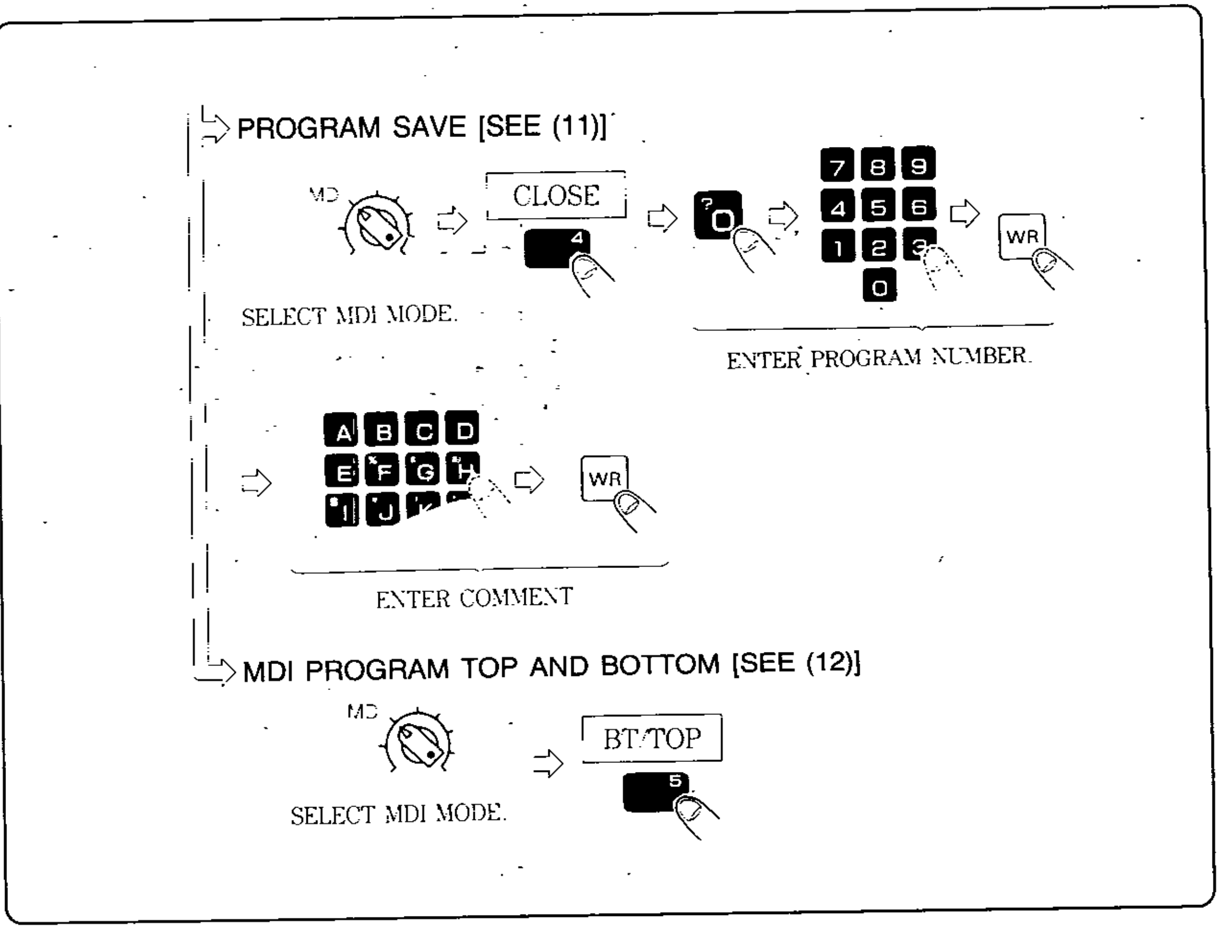
MDI PROGRAM ERASE [SEE (10)]



SELECT MDI MODE.

3

3.4 DISPLAY AND WRITE (Cont'd)



Depress the [PROG] jog soft key  to display the currently running program and information of execution.

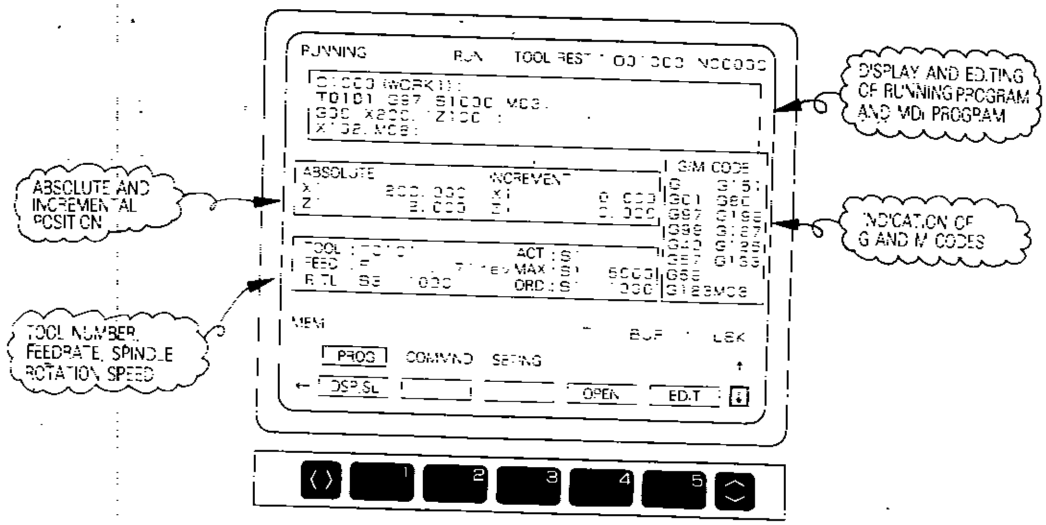


Fig. 3.4.51 Screen Displayed by RUN Key

The program screen consists of four blocks.

① **Display and editing of running program**

Display of running program is changed depending on the mode of controller operation. In the MDI mode, the MDI program is displayed. If operation is started in the tape mode, the tape program is displayed. In other modes, the contents of the part program being searched are displayed.

In the memory mode, part programs being displayed can be edited by depressing the [EDIT] function soft key. Programs that can be edited are displayed in a frame.

② **Display of command G and M code**

Modal G codes, non-modal G codes, and M codes are displayed during the auto mode operation. Modal G codes of 14 important groups among a total of 32 groups are displayed. If no non-modal G code is commanded, only G is displayed. Commanded M codes are indicated in three digits.

③ **Absolute and incremental position**

These values are convenient to check programs.

④ **Tool number, feedrate, and spindle speed**

Tool number, feedrate, spindle actual rotation speed, spindle commanded rotation speed, and spindle maximum rotation speed are displayed immediately before execution or during execution.

If a rotating tool is used, the rotation speed of the tool is also displayed.

Feedrate is indicated in number of rotations or length of travel per minute, according to feed per rotation or feed per minute.

(1) Subprogram nesting information display

Depress page key or to display the subprogram nesting information.

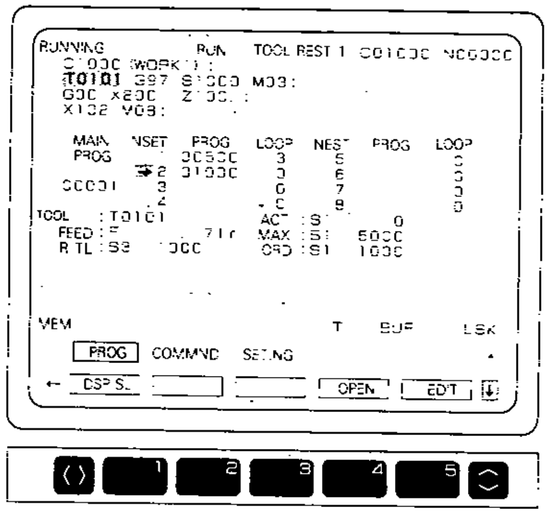


Fig. 3.4.52 Subprogram Nesting Information Screen

In **Fig. 3.4.52,** the main program number, subprogram numbers, and the number of loops (eight) are displayed. The currently running program is indicated with an arrow. Subprogram call is commanded by M98.

In this screen, macro program call is also displayed.

NOTE In the program editing status in the memory mode, the subprogram nesting information is not displayed if page key or is depressed.

Instead, the pages of the edited program are turned.

(2) **Program number search**

Program numbers can be called to execute. There are two methods:

(a) **With use of [OPEN] function soft key**

1. Select the memory mode on the machine control station.
2. Depress the [OPEN] function soft key.

"INPUT O NO:" is displayed.

3. Enter the program number after .
4. Depress the  key.

(b) **Starting with O-number input**

1. Select the memory mode on the machine control station. If program number search is attempted out of the memory mode, "MODE IS UNSUITABL" is displayed.
2. Enter the program number from the controller operator panel.
3. Depress cursor key .

The program number is searched for from the stored programs and called to the screen. The execution program number on the upper right of the screen is changed at the same time. If the specified program number is not stored in memory, a warning is issued.

Whether to call a program number that was searched for in the previous operation when power is turned on can be determined by the parameter.

Parameter pm3005 D1 0: Clear search program number at power-on.
1: Do not clear search program number at power-on.

NOTES
1. Leading zeros in a program number need not be entered.
2. In a multi-program system, search for a program number in each program.

(3) Address search

Data (character string) entered from the controller operator panel is collated with memory data (character string.). When they match, indication stops. In the memory mode, part program memory is searched. In the MDI mode, MDI program is searched.

1. Enter words to be searched for.
2. Depress cursor key or . Depress cursor key to search from the cursor location toward the beginning of data. Depress cursor key to search from the cursor location toward the beginning of data.

   During search, "SEARCHING" is displayed.
3. After search is cpmpleted, the message disappears and the cursor moves to the found word. If the word was not found a warning is displayed. In the tape mode, the warning is not displayed if operation is not restarted.
4. Cycle start after address search starts the program from the block where the cursor is.

NOTE

1. Leading zeros in data need not be specified for address search. Decimal point processing is performed during search according to the metric or inch input status. This enables search for different forms of the same addresses (such as X1. = X1000.) (Binary search function)
2. Address search cannot be specified during alarm status.

(4) Display selection

Usually, four lines of a part program are displayed for the running program indication. If necessary, up to 13 lines can be displayed.

1 Depress the [DSP.SL] function soft key.

The execution information disappears and the running program is displayed in a larger area. This function can be used even in the editing status.

The screen changes from **Fig. 3.4.53** to **Fig. 3.4.54**.

2 Depress the [DSP.SL] function soft key again to return to screen shown

in Fig. 3.4.53.

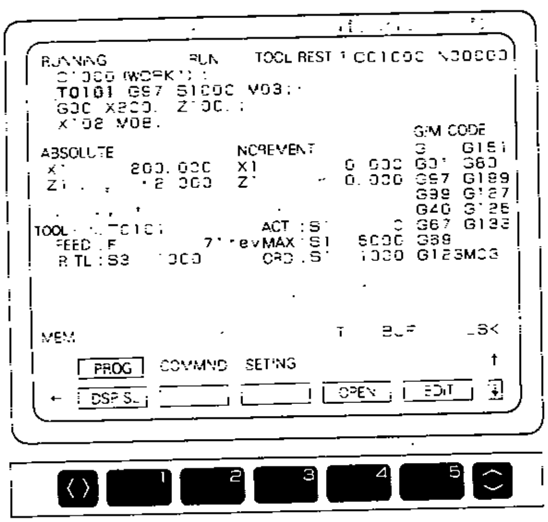


Fig. 3.4.53 Running Process Screen

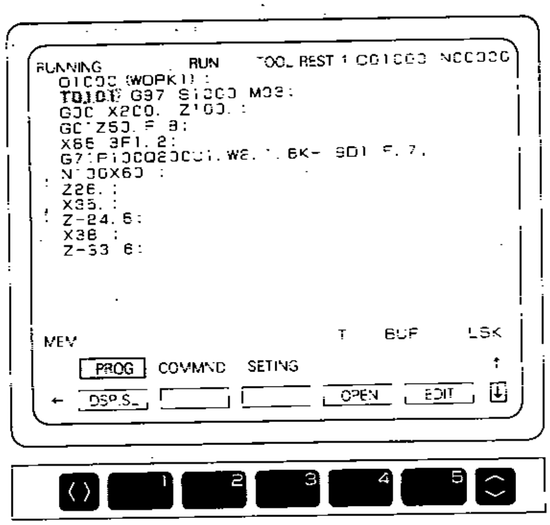


Fig. 3.4.54 Display Select Function Screen

(5) Editing part program memory

1. Depress the [EDIT] function soft key when the object part program

is selected in the memory mode. The currently selected program can be edited. When the editing status is entered, a frame appears on the program diaplay screen.

The editing function provides expansed editing functions, the same as in the [PROG] process.

2. Four lines of a part program are displayed.

Depress the [DSP.SL] function soft key to display 13 lines.

3. After editing, depress the [EDIT] function soft key again.

The part program is stored and the editing mode is terminated. While the part program is being stored, "CLOSING" is displayed.

NOTES

1. While a program is running, it cannot be edited even if the [EDIT] function soft key

is depressed. If the program is in the single-stop state, it can be edited.

2. While a program is being edited, cycle start is impossible. Use cycle start after editing is completed. Note that cycle start starts the program where the cursor was at the end of editing.

3. Address search started by the [EDIT] function soft key is the pattern search that is

performed in the [PROG] process. Leading zeros cannot be omitted.

4. The [EDIT] function soft key is not displayed in the MDI mode.

(6) Running part program

1. Select the memory mode on the machine control station.
2. Enter the program number to searh from the NC operator panel. The contents of the found part program are displayed on the screen and the cursor is highlighted. After the controller is reset, the cursor is positioned at the beginning of the program.
3. Depress the cycle start button on the machine control station. The part program is executed and the cursor moves from the beginning of one block to another as the program runs.

(7) MDI program input

An MDI program of up to 1024 characters can be entered by the following operation:

1. Select the MDI mode on the machine control station.
2. EOB is displayed in the running program display area.

   From this state, an MDI program can be written using address keys and data keys.
3. Enter an address and depress the WR key.
4. The input data are stored in the MDI buffer.

   Repeat the above operation. Up to 1024 characters can be stored in the MDI buffer. If more than 1024 characters are entered, warning message "MDI BUF OVER!" is displayed.

NOTE In a multi-program system, an MDI buffer is provided for each program. Display the screen for the program to run, and enter an MDI program to the MDI buffer of that program.

(8) Editing MDI program

An MDI program that has been entered can be edited the same as a part program stored in memory.

1. Depress cursor key ▲ or ▼ to move the cursor to the preceding or next block in the MDI program.
2. Depress cursor key ◀ or ▶ to move the cursor to the preceding or next word.

3. Depress page key  or  to turn the page of the MDI program.

4. Depress the INSRT key to insert data after the cursor location.

5. Depress the ALTER key to replace the word indicated by the cursor with an input word.

6. Depress the ERASE key to erase the word indicated by the cursor.

7. Depress the WR key to insert an input word at the end of the MDI program.

8. Whether to clear the MDI buffer by the RESET key can be determined by the following parameter:

pm3002 D1	0:	Clear MDI buffer by panel or external reset.
	1:	Do not clear MDI buffer by panel or external reset.

If the MDI buffer is not cleared by the RESET key, the cursor returns to the beginning of the MDI buffer.

NOTES

1. While an MDI program is running, it cannot be edited. If the program is in the single-stop state, it can be edited.
2. Use cycle start after editing. Note that cycle start starts the program where the cursor was when editing was completed.

(9) Running MDI program

1. Retain the MDI mode on the machine control station and depress the CYCLE START button. The input MDI program is started.

NOTE The program is started from the cursor location. The cursor must be placed at the beginning of the MDI program.

2. As the MDI program runs, the cursor moves from the beginning of one block to another.

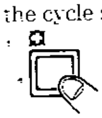
3. An MDI program can be executed out of the RUN process. Once an MDI program is entered, it can be executed on any screen.

4. Whether to erase an MDI program after its operation terminates can be determined by the following parameter:

   pm3002 D0 0: Clear the MDI buffer when MDI has terminated.

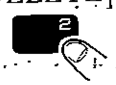
   1: Do not clear the MDI buffer when MDI has terminated if there is the end M code (M02 or M30).

NOTE In a multi-program system, each program has an MDI buffer. Input an MDI program to the MDI buffer of the program to operate, then depress the cycle start button of that program.



(10) Erasing MDI program

When the mode of the machine control station is changed to MDI, the [DELETE] function soft key is displayed. Depress the [DELETE]



function soft key to erase the MDI program stored in the MDI buffer.

This key is convenient when MDI buffer cannot be cleared by resetting from the control station because of parameter setting, or when panel reset is impossible because the MDI program is used for intervention to memory operation.

NOTE In a multi-program system, each program has an MDI buffer.
The [DELETE] function soft key deletes the MDI buffer of the displayed program.

(11) Storing MDI program

The program in the MDI buffer can be stored into memory.

1. Depress the [CLOSE] function soft key in the MDI mode.

2. "INPUT O NO." is displayed. Enter the program number to be stored.

(For example, 

)

NOTE The [CLOSE] function soft key is displayed only in the MDI mode.

3. "INPUT COMMENT" is displayed. Enter comment. The MDI program is stored with the program number and comment.

(For example, 

)

(12) MDI program top and bottom

1. Enter an MDI program in the MDI mode.
2. Depress the [BT/TOP] function soft key to move the cursor to the

last line of the MDI program.

3. Depress the [BT/TOP] function soft key again to move the cursor

to the first line of the MDI program.

3.4.7.2 Command value job (COMMND job soft key)

Command values in the currently running program can be displayed in a list.

Depress the [COMMND] job soft key.

The command value list display screen shown in **Fig. 3.4.55** is displayed. On this screen, block data being executed in the auto mode are displayed.

Conditions of Display Data

- The contents of the register under execution are displayed during automatic operation or feedhold.
- The contents of pre-reading buffer to be executed next are displayed during block stop. When the pre-reading buffer is 〈null〉, the previously executed block is displayed.

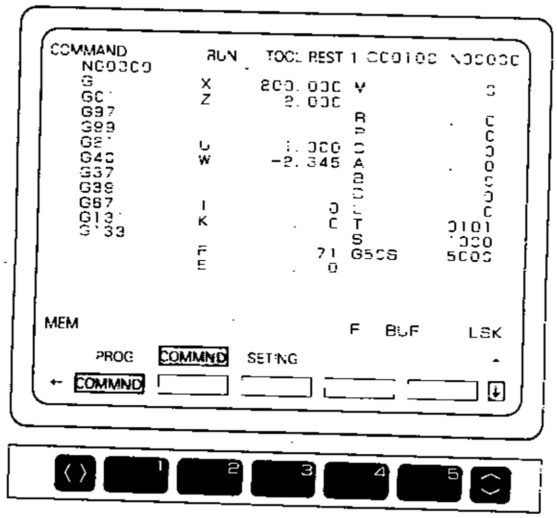


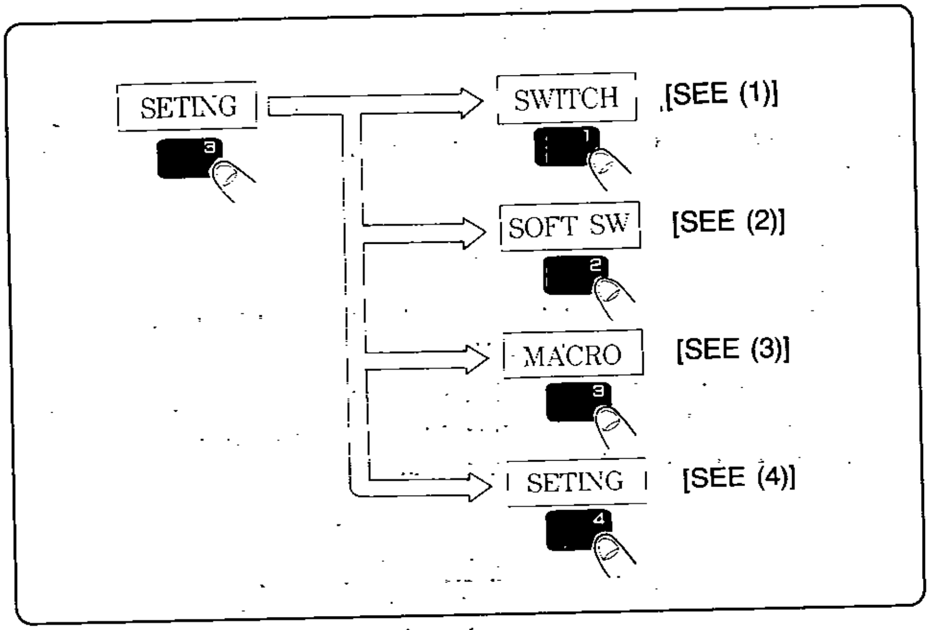
Fig. 3.4.55 Command List Display

NOTE In a multi-program system, each program has a command value screen.

3.4.7.3 Setting job

Setting data necessary for operation can be set.

KEYPOINT

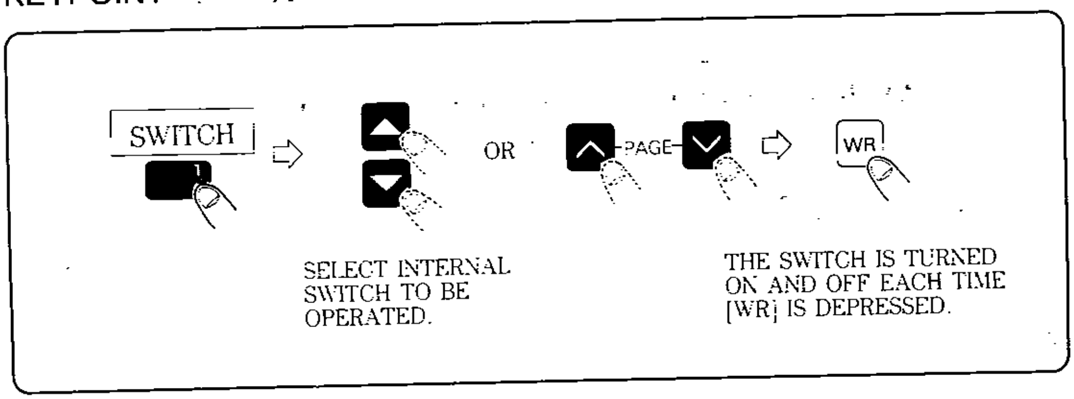


Depress [SETING] job soft key. One of the following functions is displayed:

- Internal switch function
- Software switch function
- Macro program variable function
- Setting function

(1) Internal switch function

KEYPOINT



1 Depress [SWITCH] function soft key. The internal switch screen is displayed. The switches on the screen can be turned on or off readily by setting from the NC operator panel even when switch setting is omitted on the machine control station.

There are up to 15 switches on two pages, for each program.

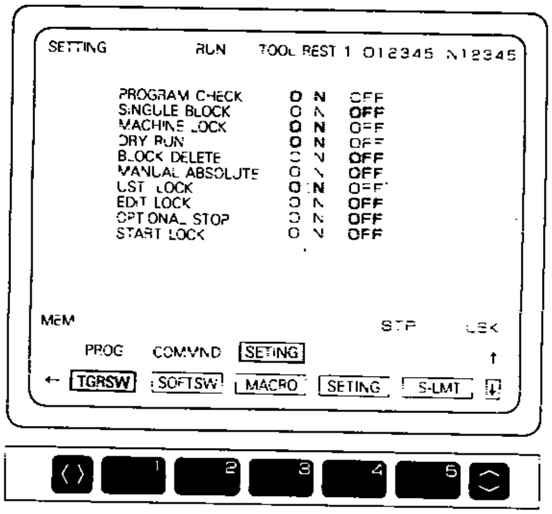


Fig. 3.4.56 Setting Display Screen (1st Page)

2 Move cursor to the internal switch to be turned on or off, then depress the WR key.

Each time the key is depressed, the switch is turned on or off. The on or off state of the switch takes effect immediately.

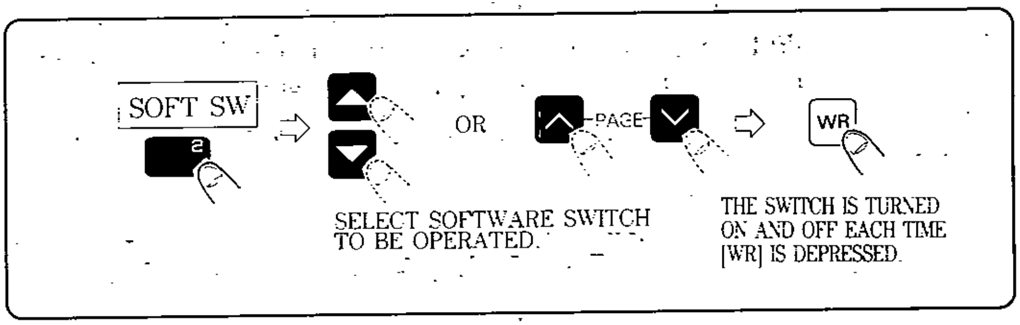
• Turning on "PROGRAM CHECK"

When "PROGRAM CHECK" is turned on, "MACHINE LOCK", "DRY RUN", and "MST.LOCK" are turned on at the same time.

This function is convenient to check a new program part before use. When "PROGRAM CHECK" is turned off, the three switches are turned off at the same time.

(2) Software switch function

KEYPOINT



Switches on the machine control station can be substituted by this screen with an option.

Depress the [SOFT SW] function soft key.

The software switch selection screen shown in **Fig. 3.4.58** is displayed. Up to 64 switches can be used, which are contained on eight pages. For more details, refer to the operation manual issued by the milling machine manufacturer.

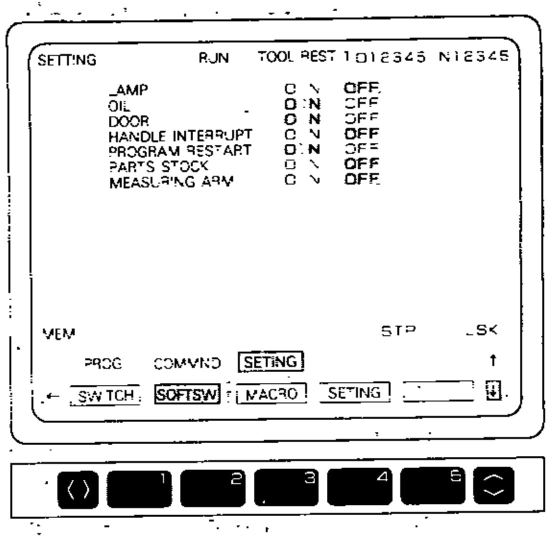
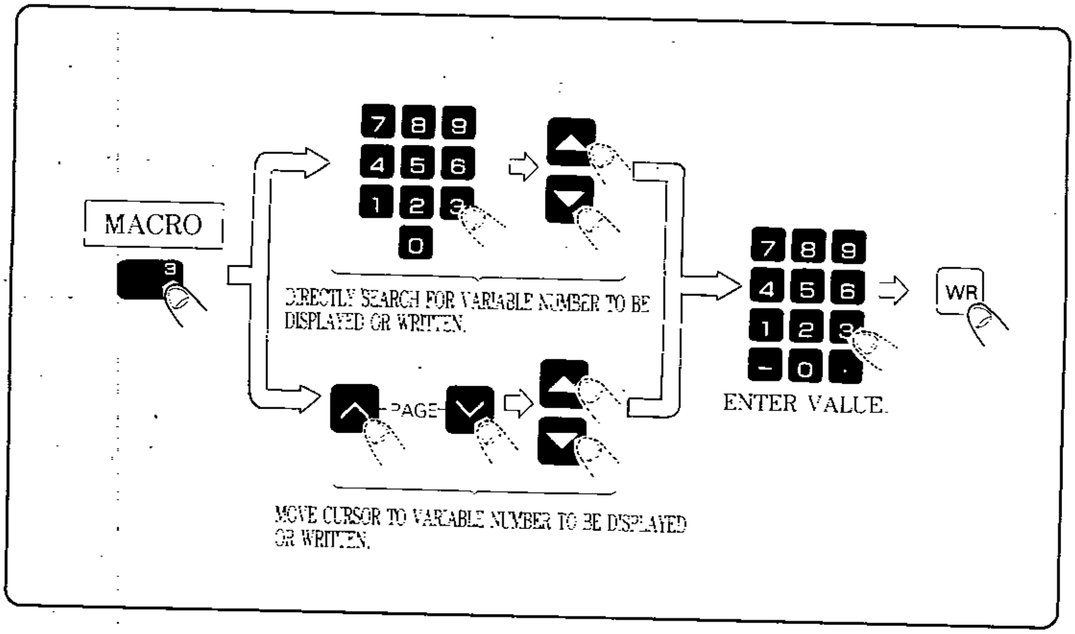


Fig. 3.4.57 Example of Software Switch Screen

(3) Macro program variable function

- Local variables (#1 to #33) and common variables (#100 to #299, #500 to #999) can be displayed and written for using macro program function option.

KEYPOINT



1 Depress the [MACRO] function soft key.

Local variables and the level of nesting are displayed.

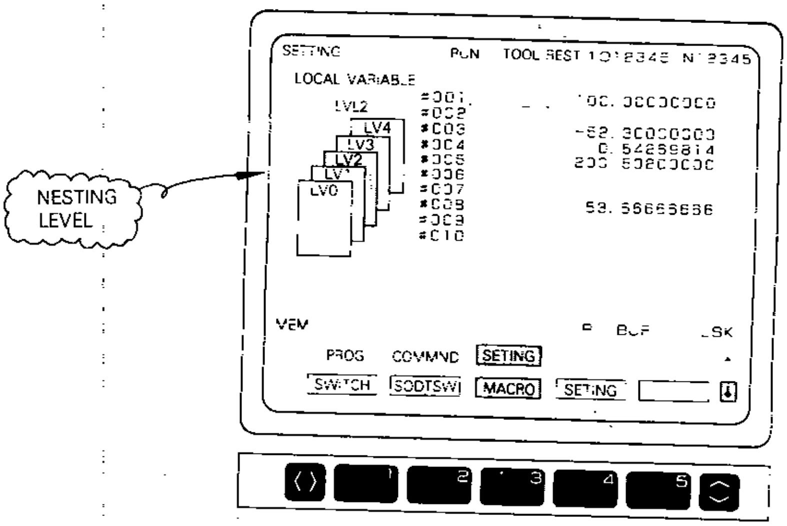


Fig. 3.4.58 Macro Local Variable Display Screen

① The level of nesting of the current running macro program is indicated from level 0 to level 4, with an illustration.

② A local variable at the currently executed level is displayed with a signed number consisting of an 8-digit integer part and a 7-digit fractional part.

③ Values of the local variables at other nesting levels cannot be displayed on the screen.

④ If data field is blank, the local variable is ⟨null⟩. After execution, all local variables are cleared to blank.

⑤ No nesting level is displayed for common variables.

⑥ In a multi-program system, each program has a local variable area independent from other programs.

Whether the common variable area is shared by the programs can be determined by the parameter:

Parameter pm4020 D1 0: In multi-program system, macro common variables are not shared by programs.

1: In multi-program system, macro common variables are shared by programs.

If this parameter is set to 1 and common variables are not shared, the variable numbers are divided according to the operation of variables sets.

2 • **Searching for variable number directly**

Enter a variable number and depress cursor key ▲ or ▼.
The variable is directly searched for.

• **Selecting variable number by cursor**

Depress page key ∧ or ∨ to turn pages of variable display.

3 Enter a value to be written, then depress the WR key.

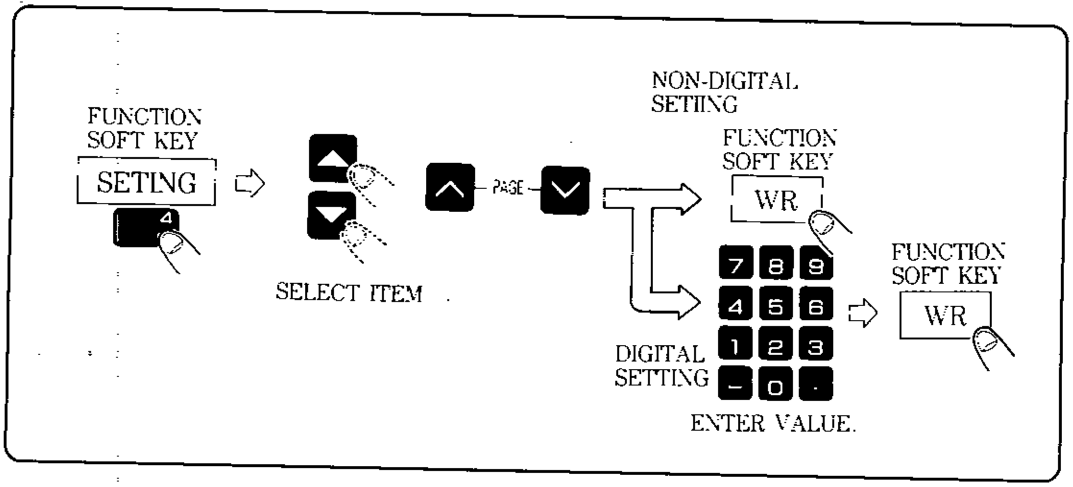
The input value is stored for the data at the variable number indicated by the cursor.

NOTES
1. Common variables can be written any time.
2. Local variables can be written only while macro is being executed. If attempted while macro is not executed, it is disregarded.

It is risky though, to change a local variable while the macro is running. Terminate operation by single-block stop, confirm that the variable can be changed safely, then rewrite the variable.

(4) Setting function

KEYPOINT



1. Depress the [SETING] function soft key. The setting screen is displayed.

2. Depress page key or or enter the page number, then depress cursor key or to display the page that contains the item to be set. Use cursor key or to move the cursor to the item to be set.

3. For a digital setting item, enter a value to be set, then depress the WR key. The set value for the item indicated by the cursor changes.

   For a non-digital setting item, depress the WR key to change the setting. The setting is turned on and off each time the key is depressed.

- **Parameter rewrite switch**

This switch is used to rewrite parameter data for [PARAM] and [SEQPRM] functions in the [MAINT] process. This switch is linked to setting data #0109.

Switch on:	Parameter rewrite enabled
off:	Parameter rewrite disabled

NOTES
1. While this switch is on, cycle start is disabled. Turn off the switch after rewriting parameters.
2. The [SETING] function soft key can be used any time regardless of the parameter rewrite status.

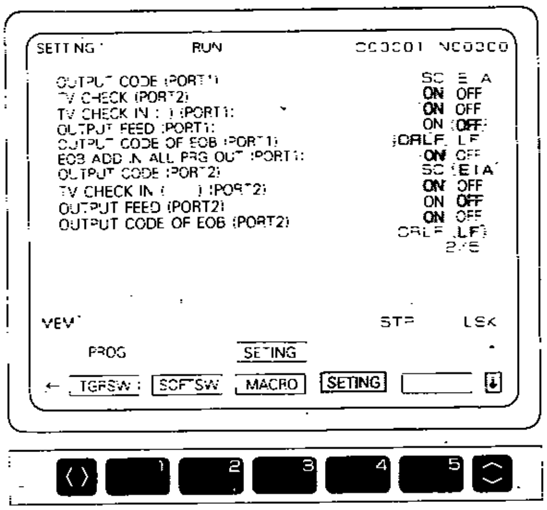


Fig. 3.4.59 Setting Screen

NOTE All items set on this screen correspond to setting data displayed by the setting function of the parameter job in the maintenance process.

For correspondence with the setting parameter numbers, refer to YASNAC i80L Operation Manual Separate Appendix (TOE-C843-11.21.)

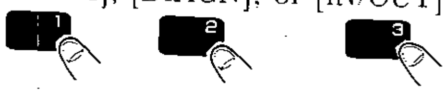
3.4.8 Maintenance Process [MAINT]

1 Depress the  key.

Any of the following jobs is displayed.

- Paramerer job → See Par. 3.4.8.1 (page 472.)
- I/O monitor job → See Par. 3.4.8.2 (page 479.)
- I/O verification job → See Par. 3.4.8.3 (page 485.)

2 To display a necessary job, depress the corresponding job soft key [PARAM], [DIAGN], or [IN/OUT].



Operations of the above jobs are explained in the following. In the maintenance process, necessary data for maintenance can be set up and displayed, or transferred to or from external equipment.

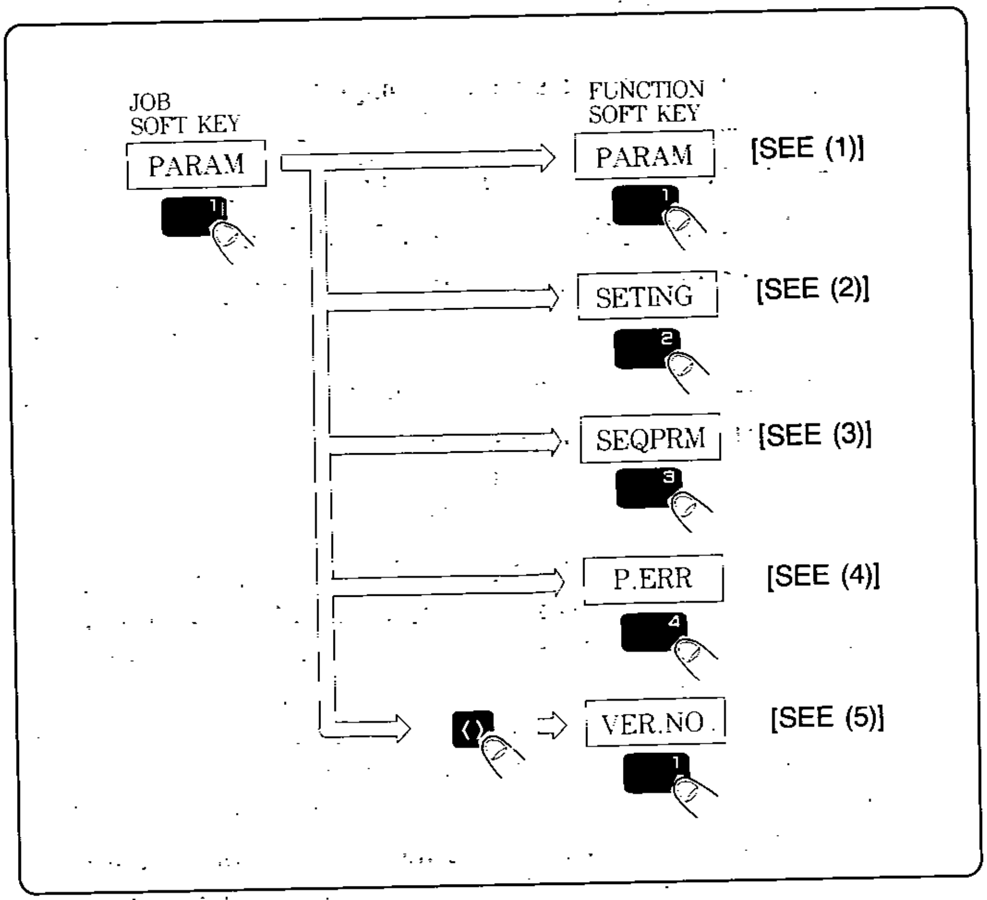
The parameter job sets up displays parameter data, setting data, keep memory data, and pitch error offset data that are contained in the controller internal memory. These data are used to turn on and off specific functions and to determine operation conditions for functions.

The I/O verification job monitors the on/off status of I/O signals among the controller, PLC, and the milling machine.

The I/O verification job transfers parameter data from the parameter job to external equipment, and vice versa. The job also verifies data by comparing input data to internal memory.

3.4.8.1 Parameter job

KEYPOINT



1 Depress the [PARAM] job soft key.

2 The [PARAM], [SETING], [SEQPRM], and [P.ERR] function soft keys are displayed. Depress  key to display the [VER.NO] function soft key.

3 Turn on the "PARAMETER CHANG" internal switch by the [SETING] function soft key of the RUN process.

This makes it possible to change parameters, keep memory, and pitch error. After setting these item, turn off the internal switch again.

(1) Parameter function

Parameter data kept in controller internal memory can be displayed and set up.

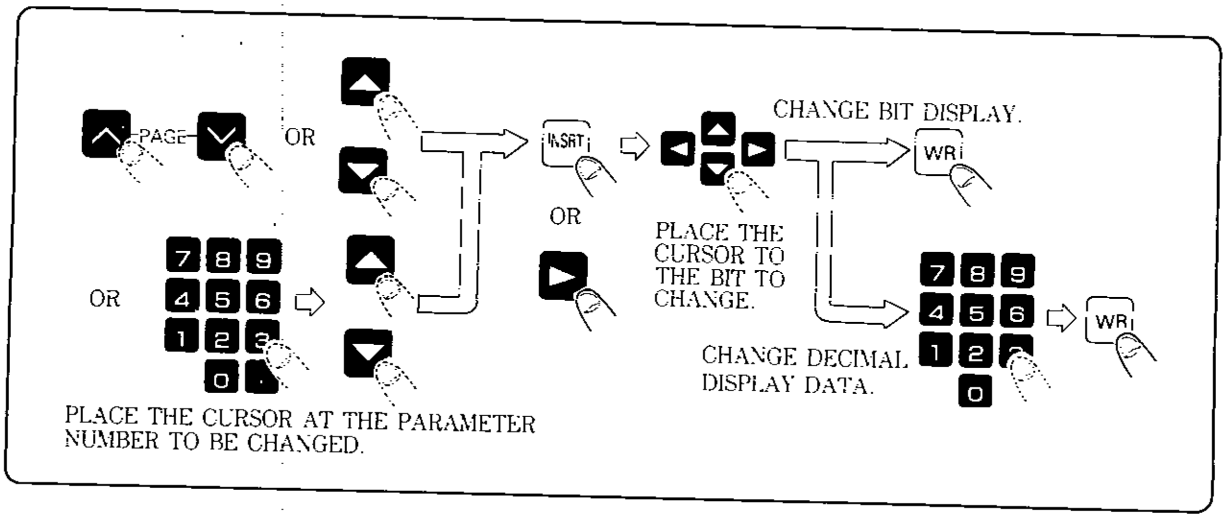
1 Depress the [PARAM] function soft key.

2 Parameters are classified into the bit type or the byte, word, and double-word type by the parameter number.

NOTE For details about the parameter number, refer to YASNAC i80L Separate Document (TOE-C843-11.21.)

⟨Bit type⟩

KEYPOINT



① Place the cursor at the parameter number to be changed, by using page key or or cursor key or ,

or by entering the parameter number directly with digit keys and depressing cursor key or .

2. Depress the INSRT key or cursor key ▶. If the INSRT key is depressed, the cursor moves to the decimal number display. If cursor ▶ key is depressed, the cursor moves to the bits.

3. Place the cursor at the bit to be changed, the depress the WR key. Each time the WR key is depressed, the bit is turned on and off alternately.

Data can be written by digit keys only when the cursor is placed at the decimal number display at the right end.

(Example) Decimal number write in bit representation

Input data	Bit 7 6 5 4 3 2 1 0	Decimal Display
0 WR	0 0 0 0 0 0 0 0	0
8 WR	0 0 0 0 1 0 0 0	8
2 5 5 WR	1 1 1 1 1 1 1 1	255

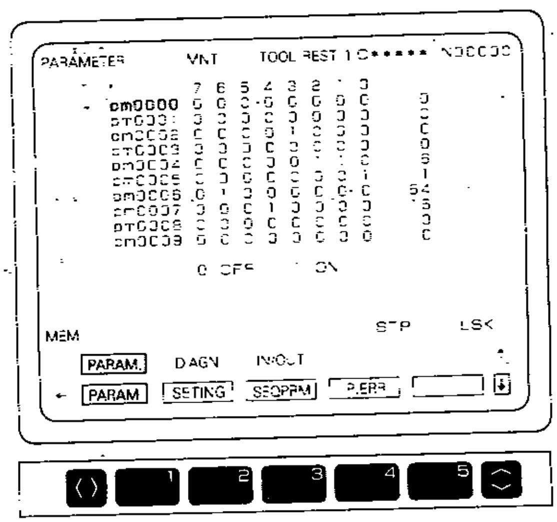
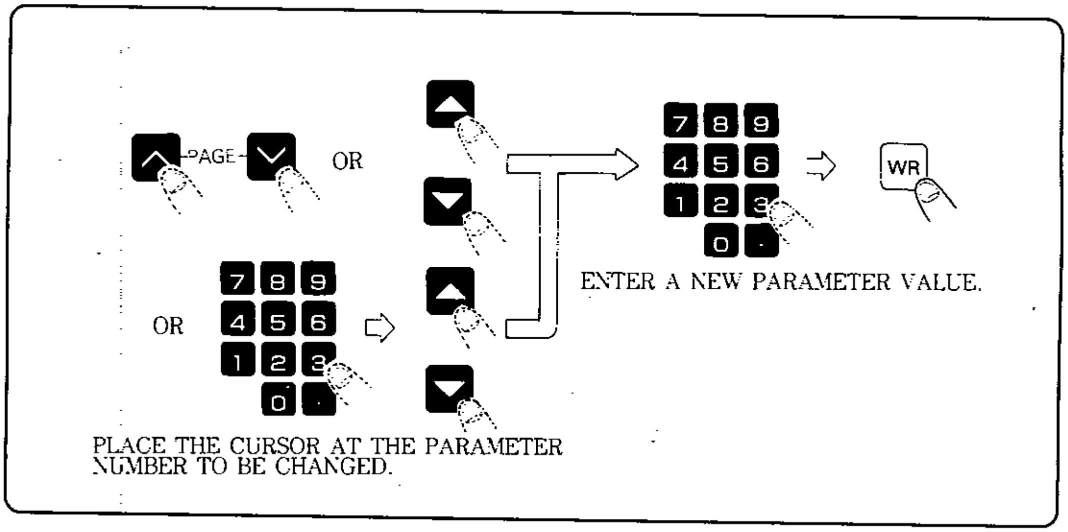


Fig. 3.4.60 Parameter Input Screen (Bit Type)

〈Byte, word, and double-word type〉

KEYPOINT



1. Place the cursor at the parameter number to be changed, by using the page key or or cursor key or , or by entering the parameter number directly with digit keys and depressing the cursor key or . To search for a parameter number, "Pm" need not be specified.
2. Enter a new parameter value.
3. Depress the WR key.

   The parameter value is entered.

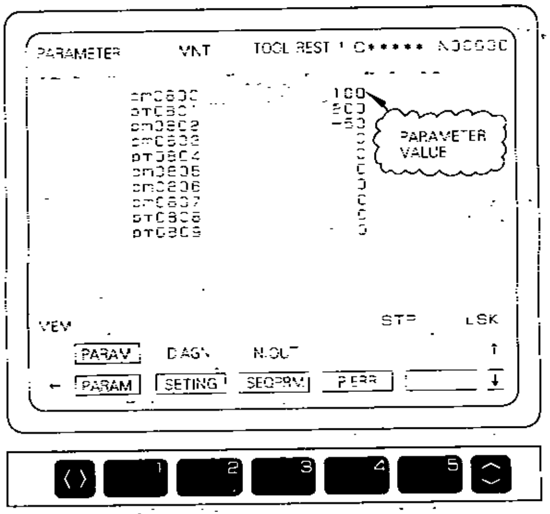


Fig. 3.4.61 Parameter Input Screen
(Byte, Word, and Double-word Type)

NOTES

1. Changed parameters take effect after the controller is reset from the panel.
2. If a parameter is written that takes effect only after power is turned off, alarm "ALM 0050: POWER OFF PRM SET" is issued. Turn off power once, then turn power on again.
3. If a parameter is written that takes effect only after the part program is initialized, alarm "ALM 0051: PROGRAM GENERATION PRM SET" is issued. If this alarm occurs, program generation is necessary. Contact your YASKAWA representative.

   Even after this alarm occurs, the system becomes operational if you turn power off and on, although the changed parameter is lost.

(2) Setting function

Setting data kept in controller internal memory can be displayed and set up.

1 Depress the [SETING] function soft key.

2 Place the cursor at the setting number to be changed.

3 Enter a new parameter value.

NOTES
1. The operation is similar to that of the parameter function, except that setting data can be changed without turning on the "PARAMETER CHANG" internal switch by the [SETING] function soft key of the RUN process.
2. Changed setting data take effect immediately.

(3) Keep memory function

Keep memory data kept in controller internal memory can be displayed and set up. The keep memory data are used by PLC.

1 Depress the [SEQPRM] function soft key.

2 Place the cursor at the keep memory number to be changed.

3 Enter a new parameter value.

NOTE The operation is the same as for the parameter function.
See **(1), "Parameter function"**.

(4) Pitch error function

Pitch error offset data kept in controller internal memory can be displayed and set up.

1 Depress the [P.ERR] function soft key.

2 Place the cursor at the pitch error number to be changed.

3 Enter a new parameter value.

NOTES
1. The operation is the same as for the parameter function.
See **(1), "Parameter function"**.
2. The pitch error function soft key is displayed only when the pitch error offset option is effective.

(5) Version number list function

The version number list screen is for maintenance of the controller, and not for use in daily operation.

To display the screen, depress key, then depress the [VER.NO] function soft key. The following items are displayed on this screen:

(a) System number

A system number is displayed under "$ SYSTEM NO" on the screen. The system number consists of the system number of the entire controller and the PC ladder system number.

(b) Option version number

An option version number is displayed at column "$ OPTION VERSION NO" on the screen. The option version number is the software version of an option board, if any.

(c) Basic version number

Basic version system numbers are displayed under "$ BASIC VERSION NO" on the screen. The basic vrsion number is the software version of a basic board. Whether there is inconsistency among the software versions of the boards is also displayed.

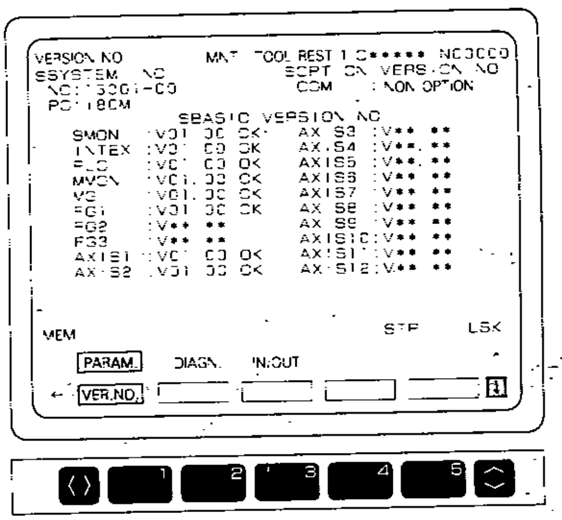


Fig. 3.4.62 Version Number List Screen

3.4.8.2 I/O monitor job

Depress the [DIAGN] job soft key to display the on/off status of the I/O

signals among the NC equipment, PLC, and machine.

NOTE The status of I/O signals can be displayed any time, including the duration of auto operation.

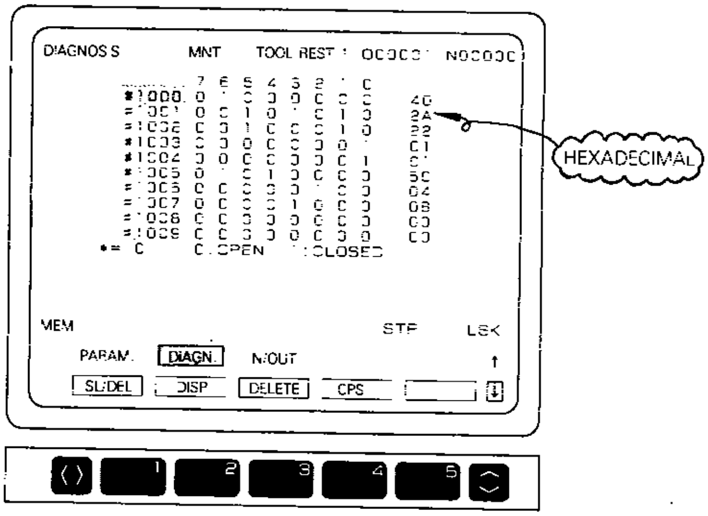


Fig. 3.4.63 I/O Monitor Screen

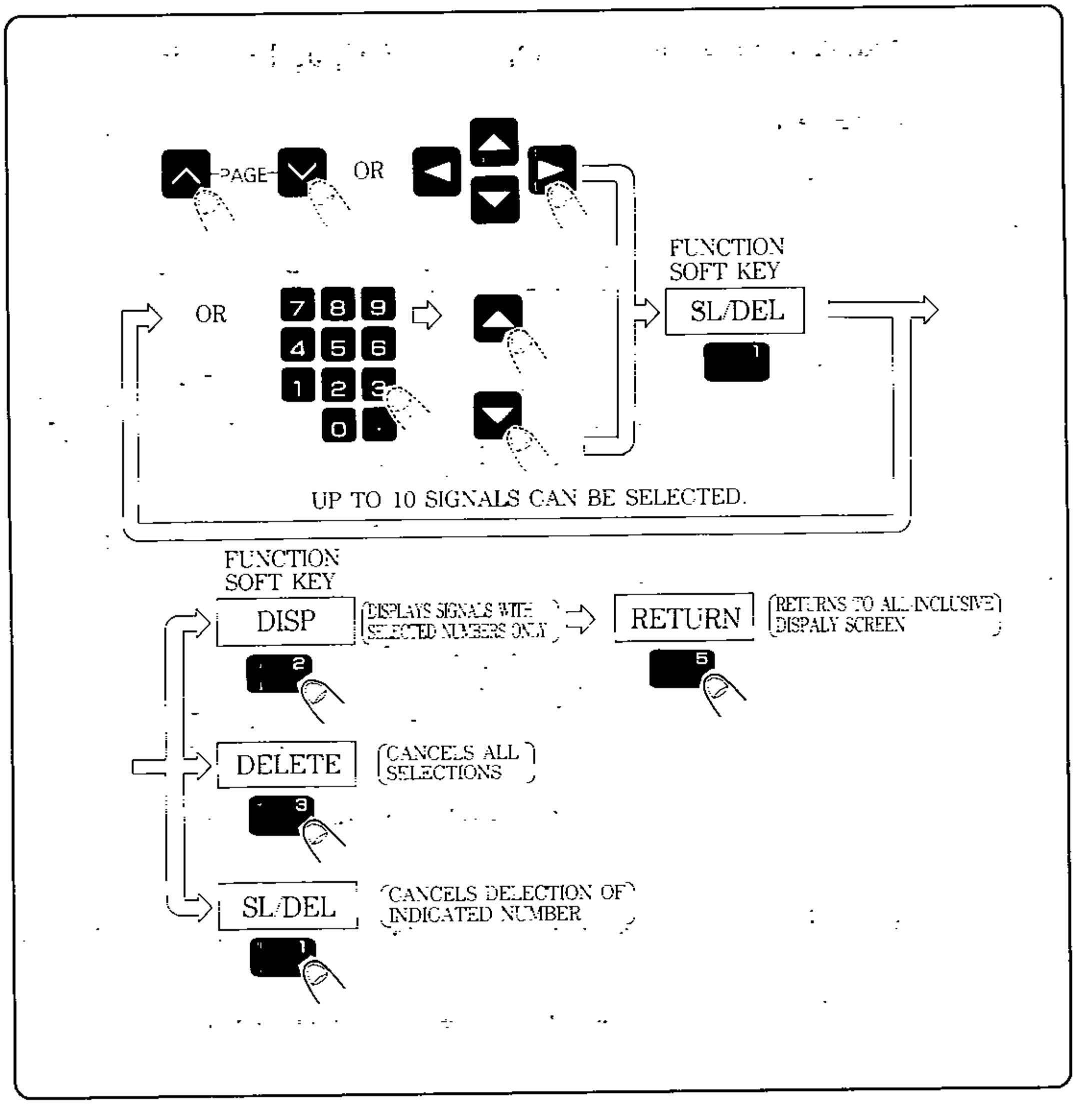
NOTES

1. For details of the display, see SECTION 4. "I/O SIGNAL DIAGNOSIS". Also refer to separate appendix (TOE-C843-11.31) "Standard I/O diagnosis Code Table" of YASNAC i80L Operation Manual.
2. The farthest value in the I/O signal status display is represented in hexadecimals, for maintenance ease.

(1) I/O signal selection and display

I/O signals can be picked up and displayed.

KEYPOINT of operation



1. Use page key , or cursor key to move the cursor to the I/O signal number to be changed.

   Otherwise, enter I/O signal number using the digit keys,

   and depress cursor key or to move the cursor to the I/O signal number to be changed.

2. Depress the [SL/DEL] function soft key. An asterisk appears to the Left of the I/O signal number, indicating that the selected number has been stored in memory. The number of I/O signal numbers is displayed on the lower part of the screen.

3. Repeat 1 and 2 to select up to 10 I/O signal numbers to be displayed.

4. ① Depress the [DISP] function soft key to display the selected numbers only.

   ② Depress the [RETURN] function soft key on the select display screen to return to the all-inclusive display screen.

   ③ Depress the [DELETE] function soft key to cancel all selections selections that have been made.

   ④ Move the cursor to an already selected I/O signal number and depress the [SL/DEL] function soft key to cancel selection of that number.

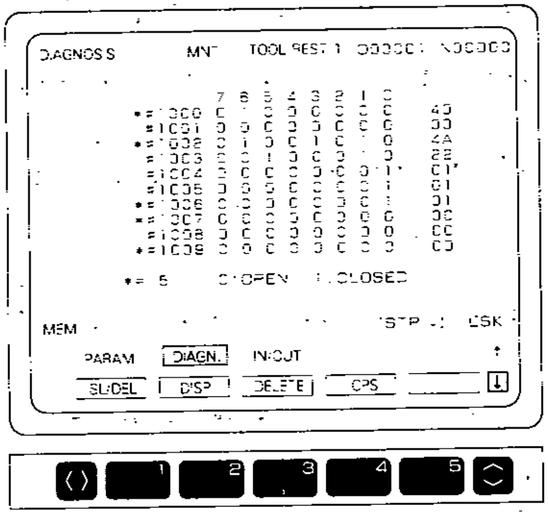


Fig. 3.4.64 I/O Monitor Screen

Depress the [DISP] function soft key to change to the following screen:

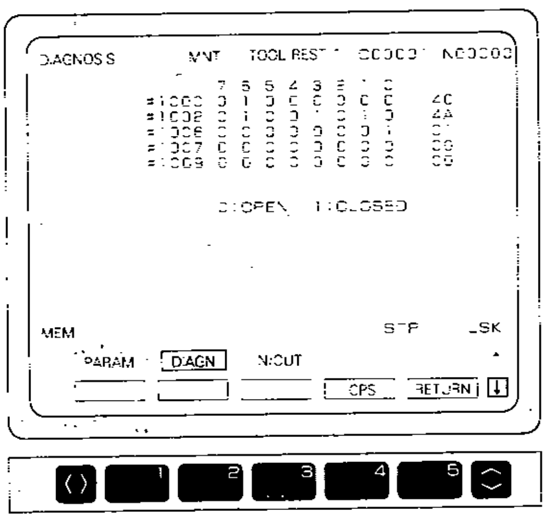


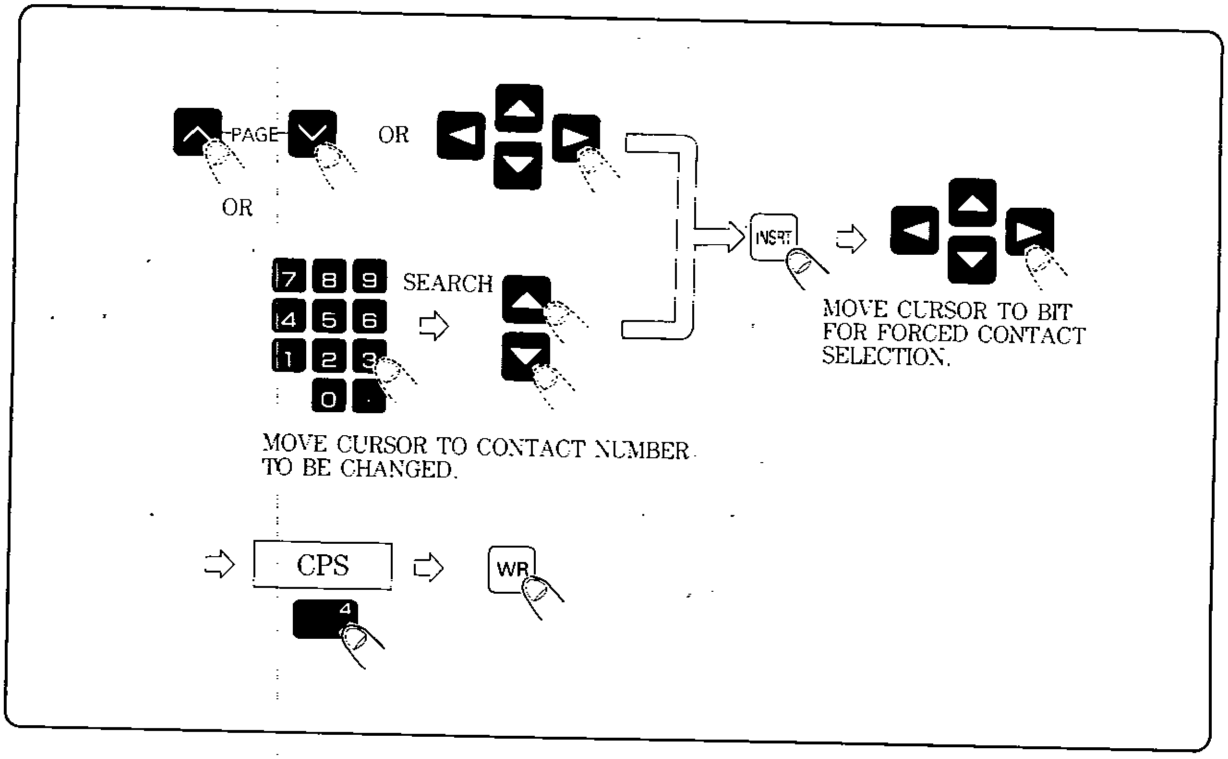
Fig. 3.4.65 Selection Display Screen

(2) Operation of forced contact selection

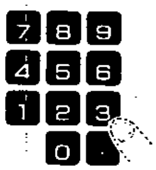
The forced contact selection function is provided to check mechanical sequences. This function can forcibly change the state of contacts without changing the mechanical sequences.

To enable this function, turn on "PARAMETER CHANG" by [SETING] function in the RUN process.

KEYPOINT



1 Use page key  or cursor key 

 or , to move the cursor to the contact number to be changed.

2. Depress the INSRT key.

   The cursor moves to the selected bit.

3. Move the cursor to the bit to select for forced contact, then depress the [CPS] function soft key. The bit is highlighted. If the function soft key is depressed again, the highlight disappears.

4. Depress the WR key. Each time the key is depressed, the bit is turned on or off.

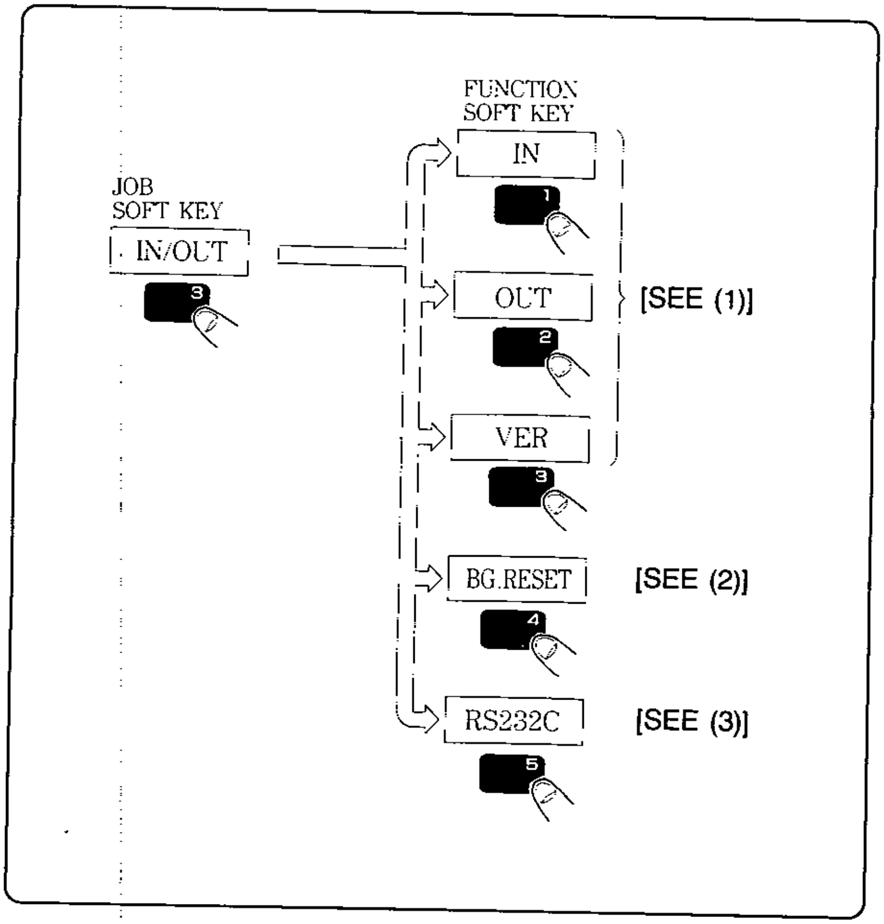
5. Move the cursor to the I/O signal number indicated by an asterisk and depress the [SL/DEL] function soft key. The forced contact selection at the number is canceled.

6. Depress the [DELETE] function soft key to cancel all forced contact selections.

3.4.8.3 I/O verification job

The I/O verification job is to input and output data between the controller and external equipment.

KEYPOINT



Depress the [IN/OUT] job soft key. The screen shown in **Fig. 3.4.66** is displayed.

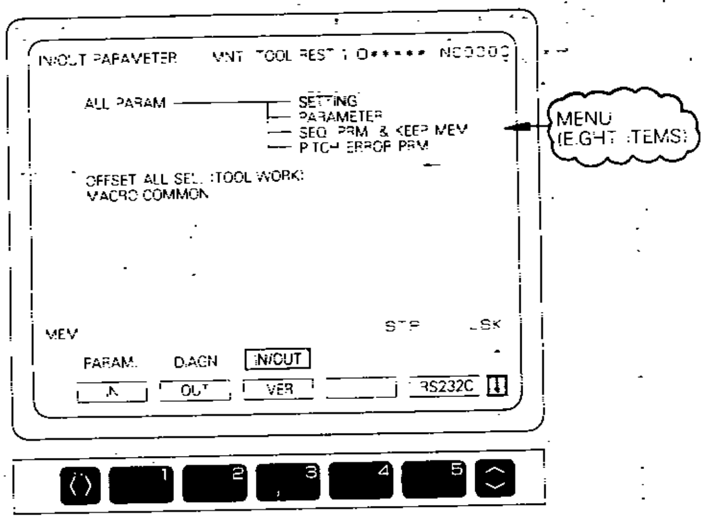
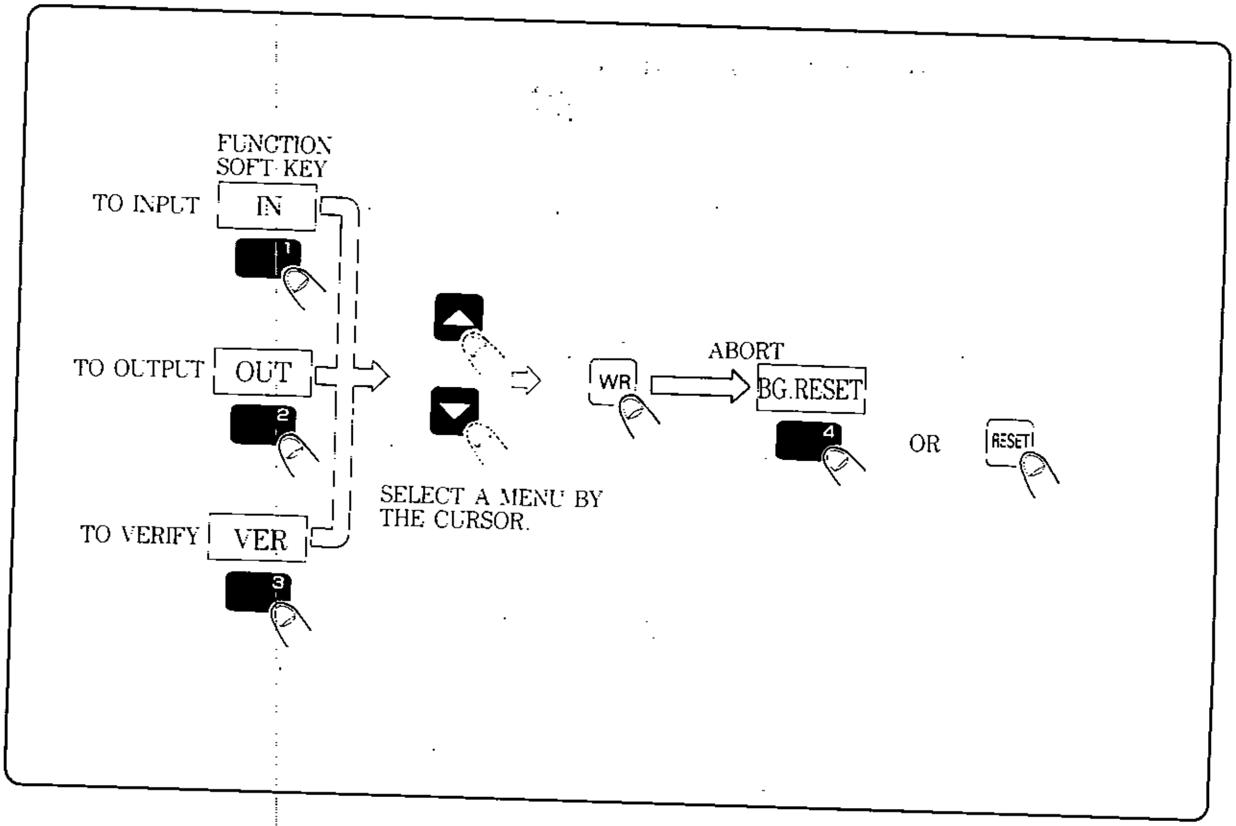


Fig. 3.4.66 I/O Verification Job Screen

(1) Input, output, and verification function

KEYPOINT



The following menus are available:

- All parameter I/O verification
  (Setting, parameter, sequencer parameter , keep memory and pitch error)
- Setting I/O verification
- Parameter I/O verification
- Sequencer parameter and keep memory I/O verification
- Pitch error offset parameter I/O verification (option)
- Offset all I/O verification
  (Tool offset, work shift)
- Tool life control, I/O verification (option)
- Macro common variable I/O verification (option)

Above data can be input, output, and verified regardless of the controller operation mode, except that in automatic operation, these operations are possible only in the stop status.

1 • To input, depress the [IN] function soft key.

• To output, depress the [OUT] function soft key.

• To verify, depress the [VER] function soft key.

2 Depress cursor key or to move the cursor to the menu to input, output, or verify, then depress the WR key.

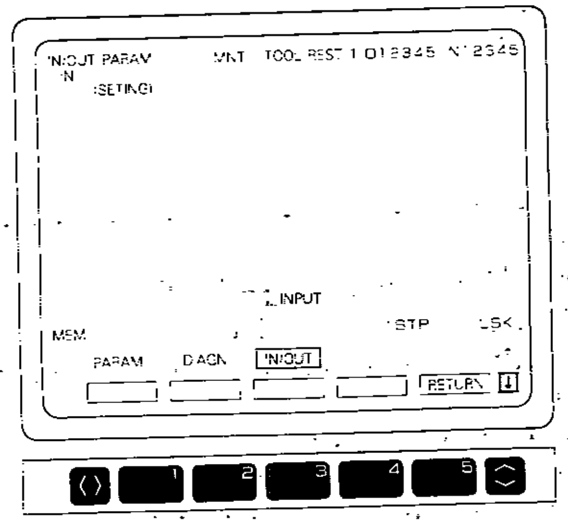


Fig. 3.4.67 Parameter Input (Output/Verification) Screen

3 ⟨Abort⟩

To abort input, output, or verification, depress the [BG.RESET] function soft key or the RESET key.

NOTE Parameter, sequencer parameter and keep memory, or pitch error offset parameter cannot be selected if internal switch "PARAMETER CHANG" of the [SETING] function in the RUN process is not turned on.

- Zero-suppress function

  Can specify whether data 0 are output at parameter output according to parameter pm0007 D6.
  pm0007 D6=0: Does not output data 0 (zero-suppress).
  pm0007 D6=1: Outputs data 0.
  Attention must be paid to the following points when the parameter output by using this function is input or verified.
- Input

  Since only data (other than 0) output by using this function are written in, write-in of 0 is not executed.
- Verify

  Since only data (other than 0) output by using this function are verified, any part which was zero at output is not verified.

(2) BG reset function

This is the background reset switch, which functions differently from the RESET switch on the operator panel.

The BG reset switch performs the following:

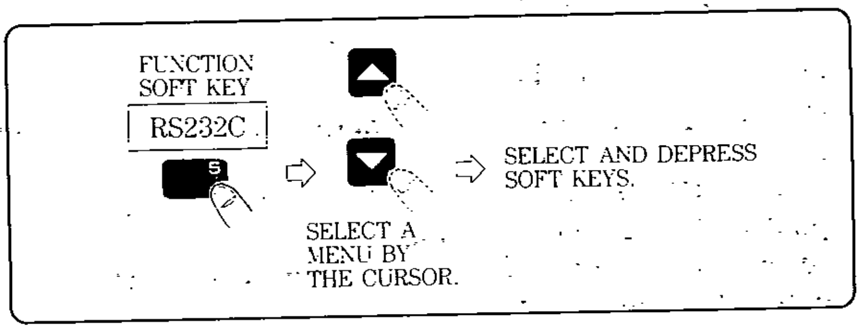
- Aborts input, output, and verification operations
- Turns on the label skip function
- Releases background alarms (ALM9000 to 9999)

The [BG.RESET] function soft key is disregarded if depressed during operation.

The reset status indication on the screen (RST) is not displayed even if the [BG.RESET] function soft key is depressed.

(3) I/O equipment setting function

KEYPOINT



Depress the [RS232C] function soft key to set up I/O equipment.

The setting procedure is the same as with the [RS232C] function soft key of the [IN/OUT] job soft key in the PROG process. Data take the same effect regardless of which screen it was set up on.

3.4.9 Common Process (COMN)

The common process deals with screens that are used in every process, not particularly in one of the editing, setup, operation, or maintenance processes explained in the previous part of this manual.

The absolute value job displays and sets up positions. The alarm job notifies an alarm if one occurs.

The time job displays and sets up date and time with the calender function. Work times can be displayed and reset during operation.

1 Depress the 

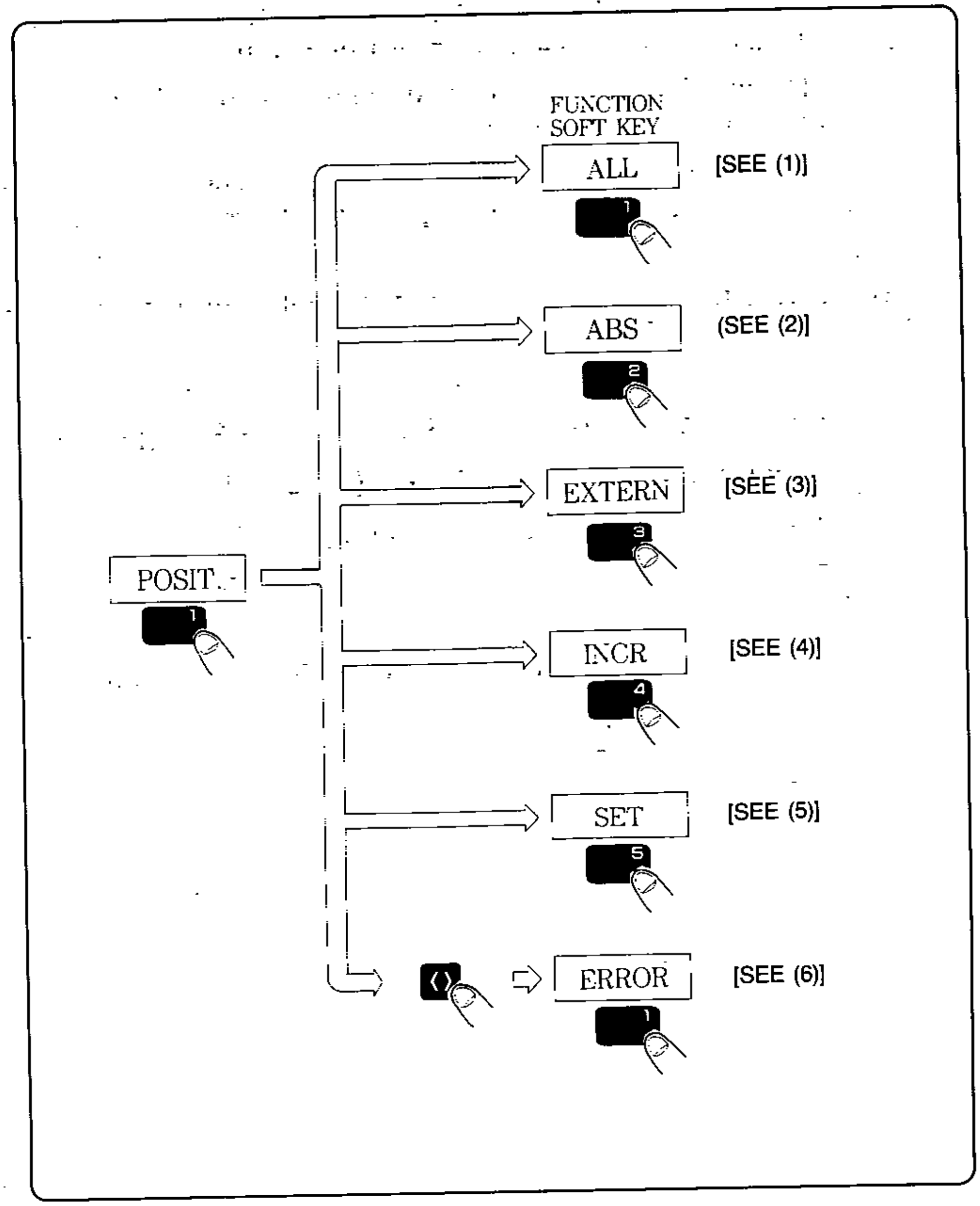
key. Any of the following jobs is displayed.

> • Absolute position job → See Par. 3.4.9.1 (page 493.)
> • Alarm job → See Par. 3.4.9.2 (page 501.)
> • Time job → See Par. 3.4.9.3 (page 505.)

2 To display a necessary job, depress the corresponding job soft key ([POSIT.], [ALARM], or [TIME].)

3.4.9.1 Absolute position job (POS. job soft key)

KEYPOINT



Depress the [ABS.] job soft key. Any of the following functions is displayed:

- All absolute position function
- Absolute position function
- External absolute position function
- Incremental function
- Error pulse function

(1) All absolute position function

Depress the [ALL] function soft key to display the all absolute position

display screen.

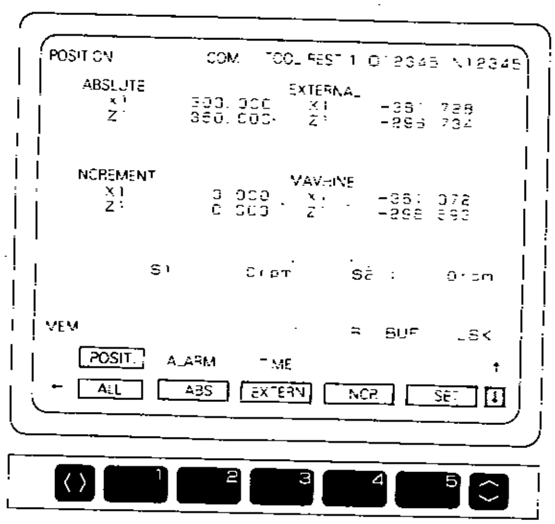


Fig. 3.4.68 All Absolute Position Display Screen

Display of absolute positions are explained in the following:

(a) Absolute position

An absolute position value indicates the current position of a tool in a specified coordinate system.

Whether to add tool offset to absolute position indication can be detemined by the parameter:

Parameter pm3000 D2 0: Add tool position offset to absolute position indication.
1: Do not add tool position offset to absolute position indication.

To preset absolute position, use [SET] key [See (5)] of the [POSIT.] function.

(b) External absolute position function

An external absolute position indicates an accumulated length of travel of a tool from a specific position preset by [SET].

(c) Incermental value

An incremental value indicates the following:

- In the auto mode, a momentary distance from the current position of a tool to the end point of that block.
- In the manual mode, the distance from the current position of a tool to the start point of manual operation. This indication is reset the 0 when the auto mode is restarted.

(d) Machine coordinate system

A value in the machine coordinate system is the current position of a tool in a coordinate system that has the origin at the referent point to which the tool returns.

NOTES
1. The axis name of each absolute value is the axis name if indication from pm1100. If this parameter is not set, the default axis name of the controller is used. In this case, the axis name may be varied be the number of programs.
2. If total number of axes of all programs is six or less, all axes are displayed at the same time. If more axes are used, one screen displays axes of a single program.

(2) Absolute value

Depress the [ABS] function soft key to display an absolute position.

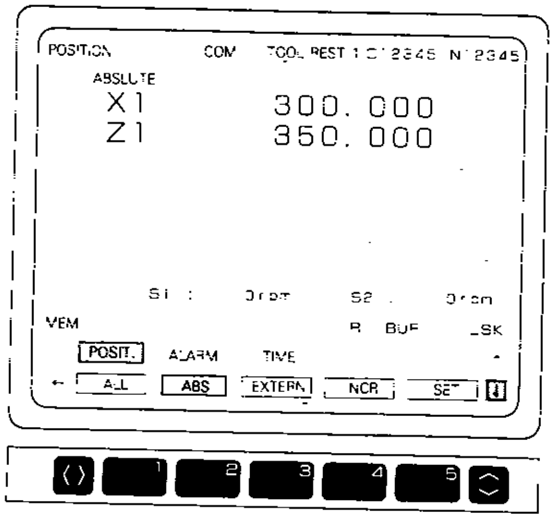


Fig. 3.4.69 Absolute Position Display Screen

(3) External absolute value

Depress the [EXTERN] function soft key to display an external absolute position.

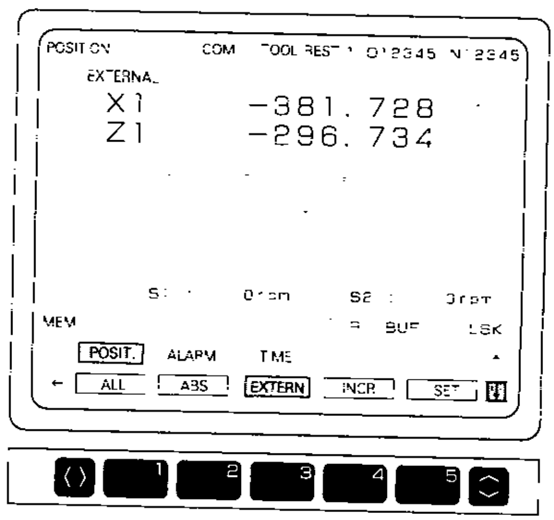


Fig. 3.4.70 External Absolute Position Display Screen

(4) Incremental value

Depress the [INCR] function soft key to display an incremental value.

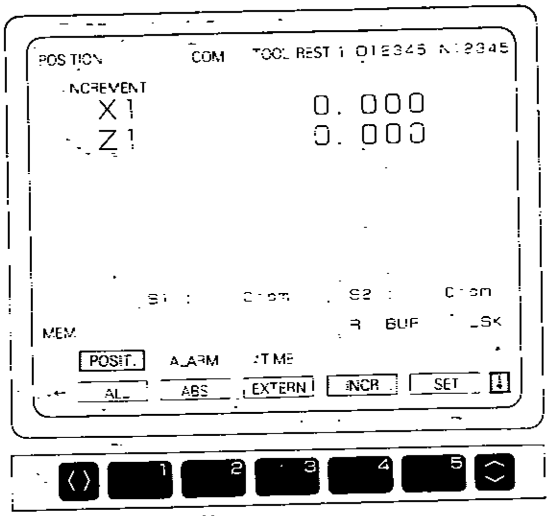


Fig. 3.4.71 Incremental Value Display Screen

(5) Coordinate system setup

A coordinate system can be set up when [ALL], [ABS], or [EXTERN] is selected.

(a) When the [ALL] function is selected

With the [ALL] function, external absolute values can be used to set up a coordinate system.

Setting of a coordinate system with external absolute values is always effective even during motion of axes by automatic operation by part program.

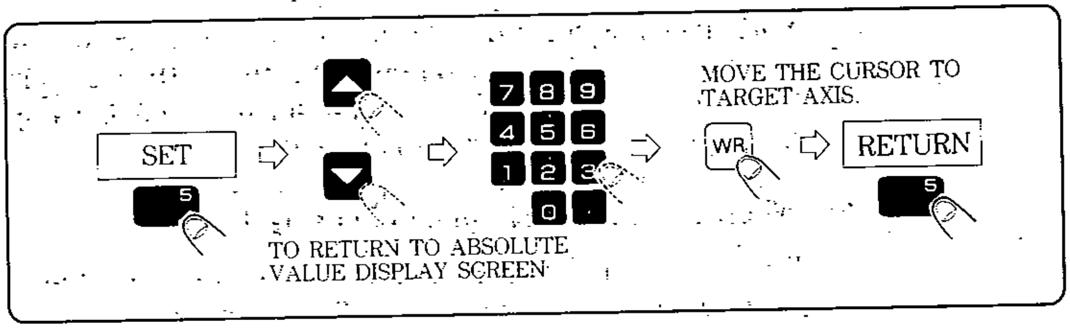
(b) When the [ABS] function is selected

With the [ABS] function, absolute values can be used to set up a coordinate system. This operation is effective in the manual mode (rapid traverse, step, or handle.) If a coordinate system is set up with [ABS] in other modes, a warning is issued.

(c) When the [EXTERN] function is selected

With the [EXTERN] function, external absolute values can be used to set up a coordinate system. Setting of a coordinate system with external absolute values is always effective even during motion of axes by automatic operation by part program.

KEYPOINT



1 Depress [SET]. The name of the axis to be set up is displayed in reverse. 

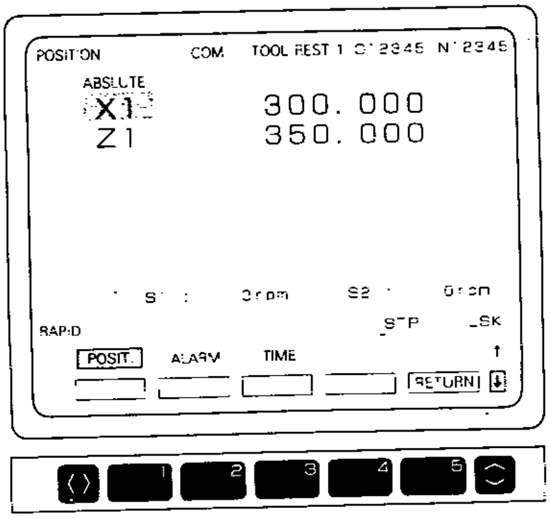
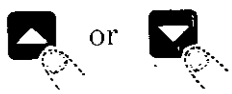


Fig. 3.4.72 Coordinate System Set-up Screen (Absolute Value)

2 Depress cursor key  to move the cursor to the axis to be set up.

3 Enter the coordinate system setup value.

(For example, )

The coordinate system is set up and the cursor automatically moves to the next axis.

4 Depress the [RETURN] function soft key to return to the absolute value  display screen.

NOTES 1. To set 0 for the coordinate setup value, just depress the  key.

2. If a coordinate system is set up in a mode where it is impossible, a warning is issued.

(6) Error pulse display function

Depress the [ERROR] function soft key. The error pulse diaplay screen appears.

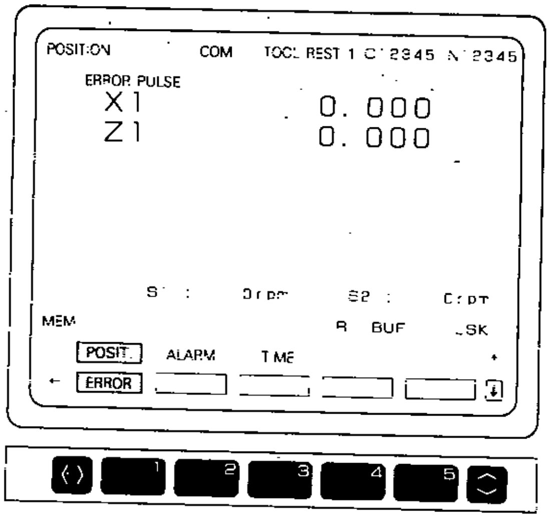
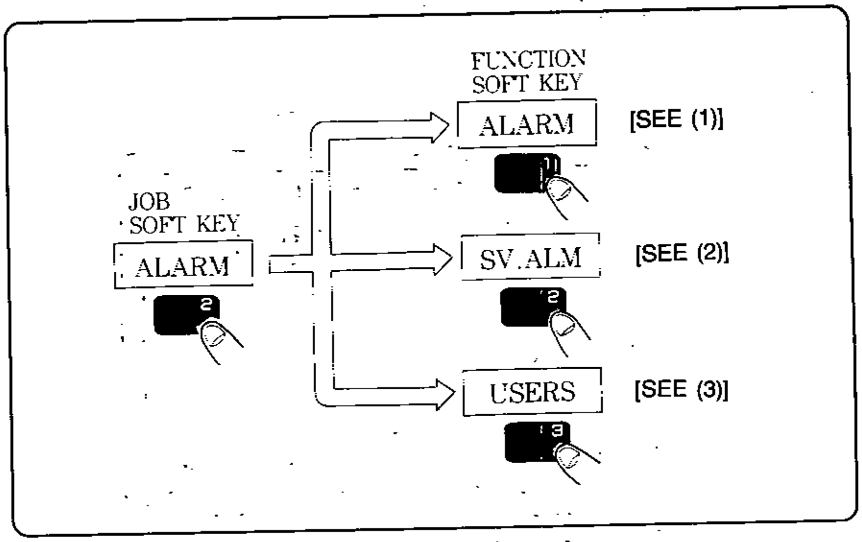


Fig. 3.4.73 Error Pulse Display Screen

The error pulse display function displays the contents of the command pulse accumulation register in the controller. The command pulse accumulation register is 0 when power is turned on, and accumulates all command pulses until power is turned off.

3.4.9.2 Alarm job (ALARM job soft key)

KEYPOINT



Depress the [ALARM] job soft key. The alarm function or the users

message function is displayed.

(1) Alarm function

If an alarm occurs in the controller, regardless of the mode or process, "ALM", "BGA" (background alarm), "A/B", or "B/B" (means that the battery alarm has also occurred) is displayed blinking in the alarm status field on the screen. Also, the highest priority alarm number, message, and the error program are displayed in the alarm message field.

Depress the [ALARM] function soft key to display the alarm screen

for the entire system to monitor all current alarms.

Alarm numbers and the corresponding alarm messages are displayed in the order of the priority for each program.

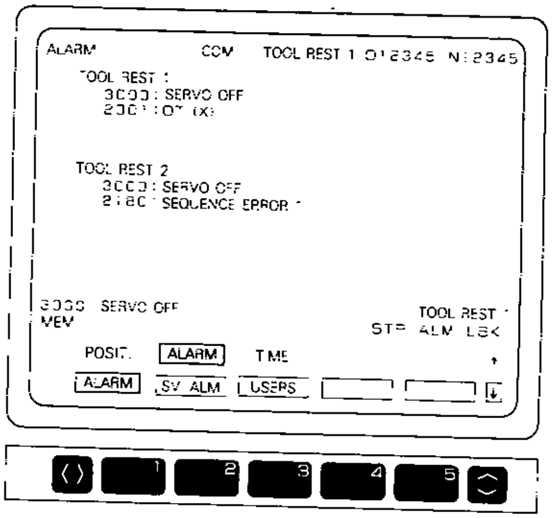


Fig. 3.4.74 Alarm Function Screen

(2) Servo alarm function

If a servo alarm of alarm number 3101 to 3108 occurs, this function displays the details.

- Depress the [SV.ALM] function soft key to display the servo alarm detail names.

- Alarms are displayed in ascending order of the alarm number. If more than one page is used, press page key or to turn pages.

- There are eight alarms: overcurrent, MCCB trip, regeneration error, overvoltage, insufficient voltage, heat sink overheat, current command cable break, and missing phase.

For details of the alarms, refer to the separate appendix of YASNAC i80L Operation Manual and YASNAC i80 Maintenance Manual (TOE-C843-11.2.)

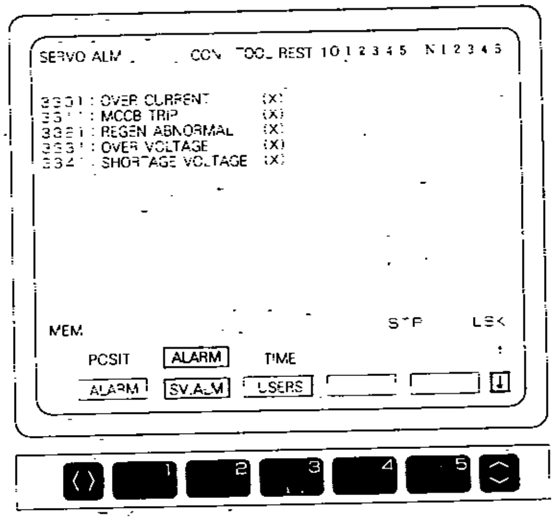


Fig. 3.4.75 Servo Alarm Function Screen

(3) Uses message function

This function displays messages on the CRT screen using the machine sequence controller (PLC system.)

Usually this function is used to display the alarm causes detected by the PLC. It can be used simply for message display. When a message display command (macro command) is executed in the PLC, "ALM" or "A/B" is displayed blinking in the alarm status field on the screen.

If the [ALARM] function soft key is depressed when "ALM" or

"A/B" is displayed, the following appears:

ALM	1080	SEQUENCE ERROR 0
ALM	2180	SEQUENCE ERROR 1
ALM	3240	SEQUENCE ERROR 2

Depress the [USERS] function soft key next to display the users

message screen to see detailed error codes sent from the PLC:

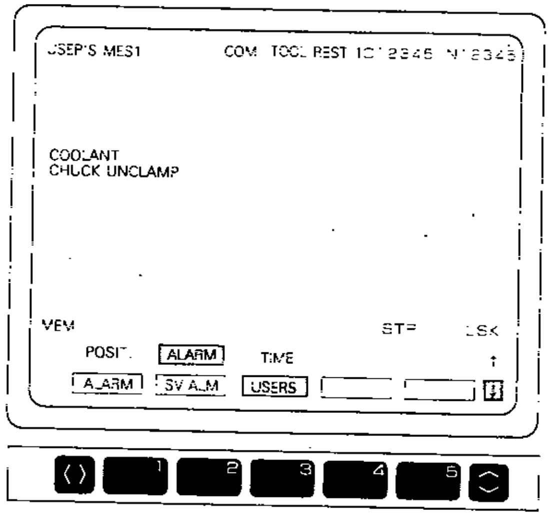
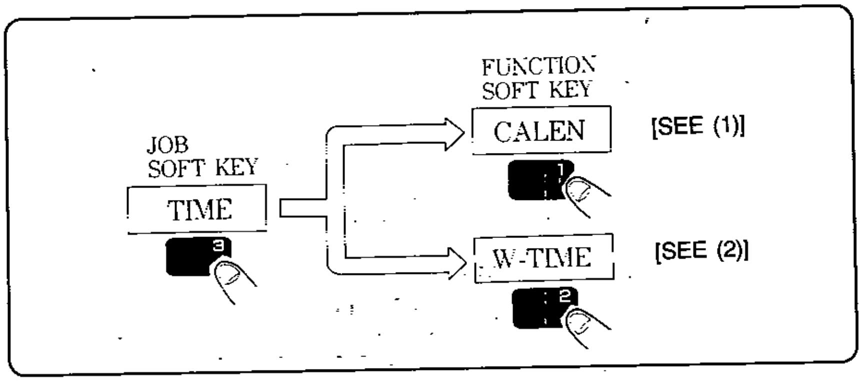


Fig. 3.4.76 Users Function Screen

NOTE For more details, refer to the manual issued by the milling machine manufacturer.

3.4.9.3 Time job (TIME job soft key)

KEYPOINT



1. Depress the [TIME] job soft key.

The calendar screen or the work time screen appears.

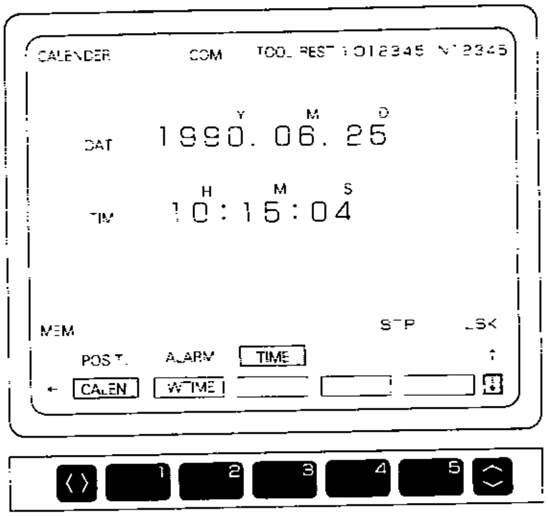
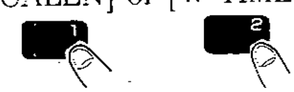


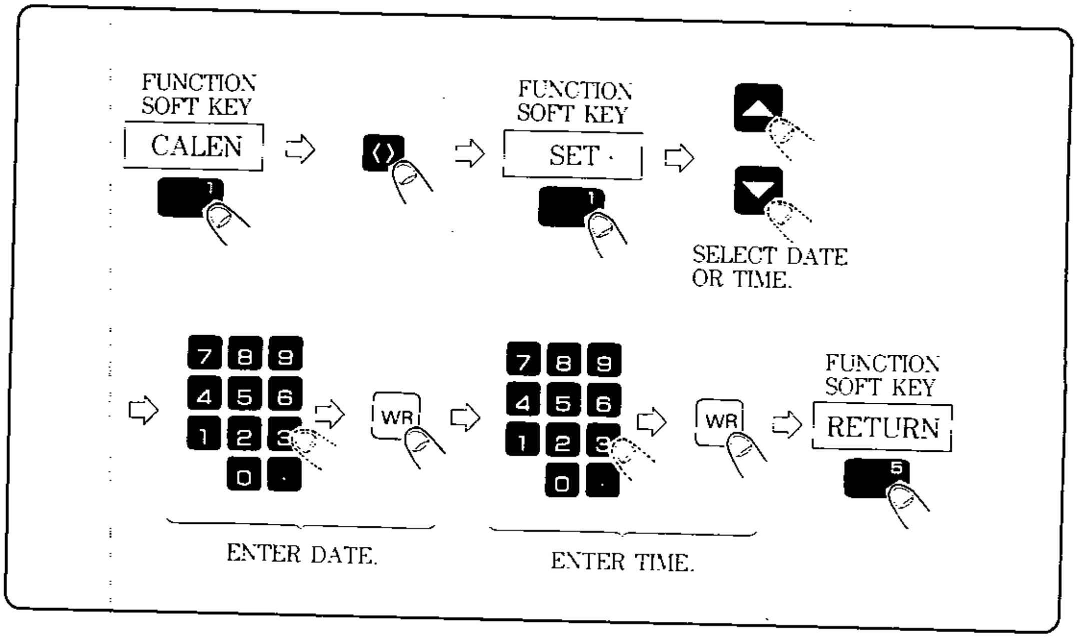
Fig. 3.4.77 Calendar Screen

2. Select the [CALEN] or [W-TIME] function soft key.



(1) Calendar function

KEYPOINT



■ Depress the [CALEN] function soft key. The date and time held by

the calendar function is displayed.

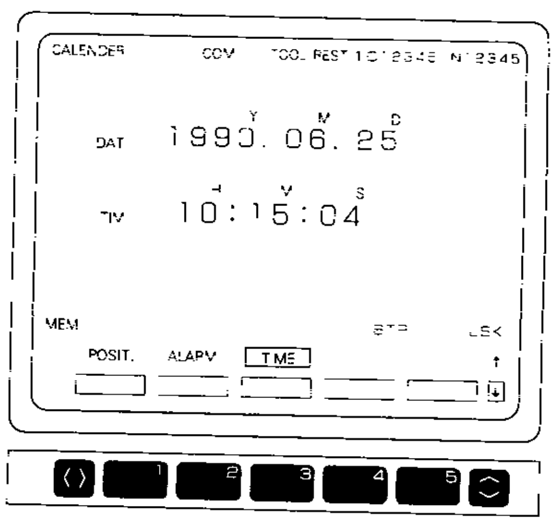


Fig. 3.4.78 Calendar Setup Screen

2 To set date and time, depress ⟨⟩ key.

The [SET] function soft key is displayed.

3 Depress the [SET] function soft key.

The key is highlighted and the cursor is displayed at the date field. The setup mode has started.

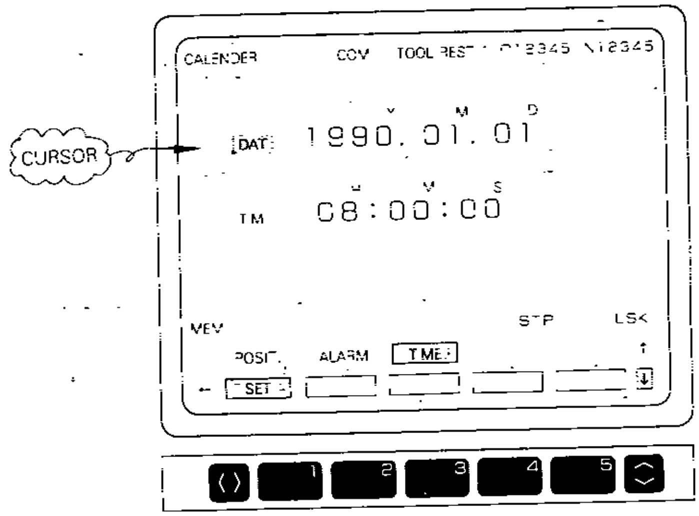


Fig. 3.4.79

4 Select date or time using cursor key ▲ or ▼.

5 Enter date.

(For example, 1 9 9 1 . 0 1 . 0 1 WR)

6 Enter time.

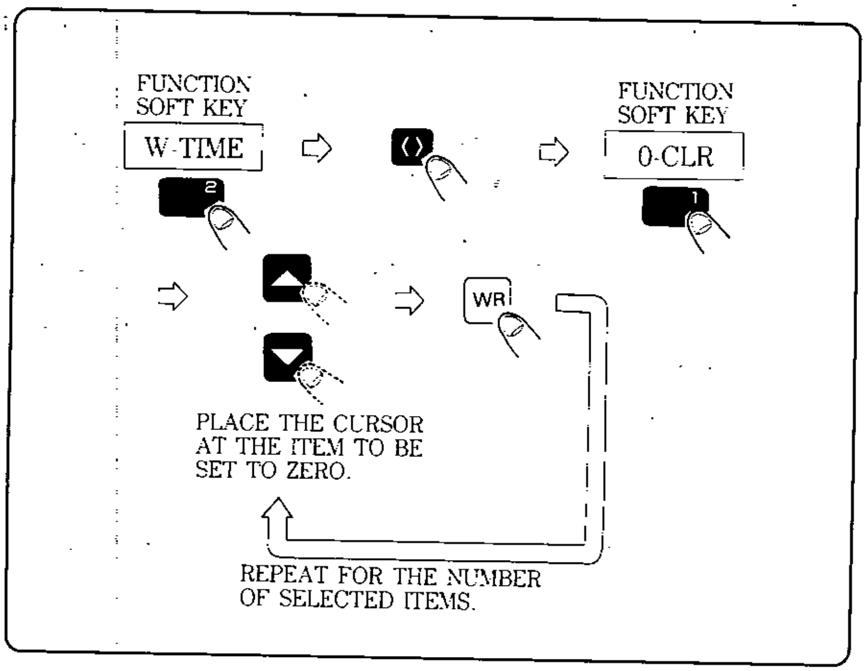
(For example, 1 2 . 0 0 . 0 0 WR)

7 Depress the [SET] function soft key.

The cursor disappears and the entered date and time are displayed.

(2) Work time function

KEYPOINT



1 Depress the [W-TIME] function soft key.

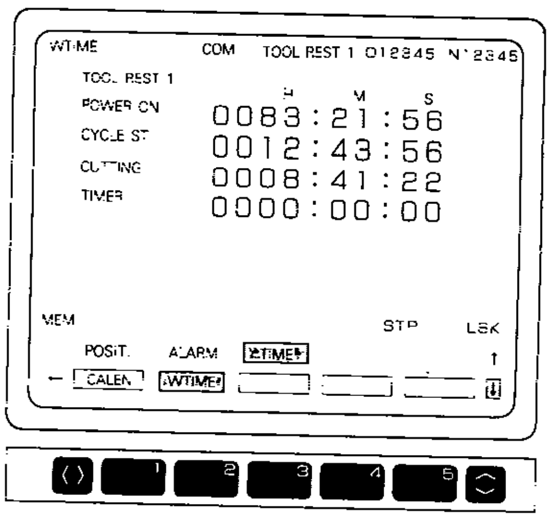


Fig. 3.4.80 Work Time Screen

The NC controller accumulates work times internally. Use this function to monitor the processing time of a single workpiece or operation duration of a milling machine.

There are four types of work times, each counted in hours, minutes, and seconds:

(1) Power on .... Total time after power-on

(2) Cycle start .... Accumulated time of total duration of automatic operation

(3) Cutting feed ..... Accumulated time of total duration of cutting.

(4) External timer .. Accumulated time of total duration of external input signal being on. (option)

In a multi-program system, each program has independent timers except for the power on timer.

2 To reset a work time, depress the operation change key ⟨⟩. The [0-CLR] function soft key is displayed.

3 Depress the [0 CLR] function soft key.

The key is highlighted and the cursor is displayed at the power-on field. The zero clear mode is entered.

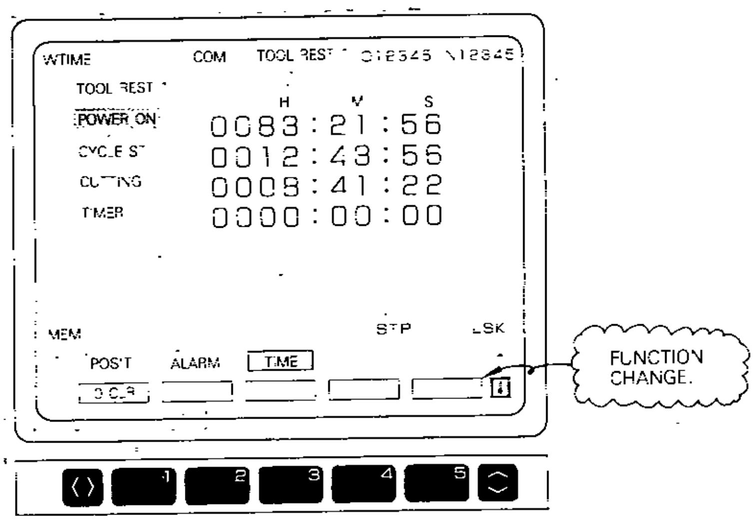


Fig. 3.4.81 Zero Clear Mode Screen

4 To reset a work time to zero, place the cursor to that work time using cursor key ▲ or ▼, then depress the WR key.

The work time at the cursor is cleared to zero. The cursor moves to the next item.

5 Depress the [0 CLR] function soft key.

The cursor disappears and the zero clear mode is canceled.

3.4.10 Power-on initial indication

An arbitrary initial screen can be displayed after power-on with an option. Specify the characters to be displayed, size, and position of the characters using the parameters:

(a) Characters

Parameter pm3400-3409: 1st line on power-on title
Parameter pm3410-3419: 2nd line on power-on title

Up to 10 ASCII characters can be set per line with these parameters. ASCII codes from 20h "SP" to 5Fh "–" can be used. If the title is shorter than 10 characters, set 0 for the end code.

(Example) YASKAWA = 59h, 41h, 53h, 4Bh, 41h, 57h, 41h, 0 (haxadecimal)
= 89, 65, 83, 75, 65, 87, 65, 0 (decimal)

(b) Display size

Parameter pm3420 Display character size
Size of characters on the 1st line Place on the 1st line. (Range: 1 to 3)
Size of characters on the 2nd line Place on the 2nd line. (Range: 1 to 3)

Character size can be set for these parameters as follows:

(Example) Character on the 1st line in × 2 scale, character on the 2nd line in × 1 scale
Set 12h (haxadecimal) = 18 (decimal)

(c) Display start position

Parameter pm3421 Display start postion in horizontal direction

1st line start position	Place in the 1st or 2nd column (Range: 1 to 39)
2nd line start position	Place in the 3rd or 4th column (Range: 1 to 39)

This parameter determines the display start position in the horizontal direction. The vertical direction is fixed to the system.

(Example) When 10 characters are on the 1st line and 12 characters are on the 2nd line, set as follows:
Set 0C0Ah (hexadecimal)=3082 (decimal).

YASNAC i80L
INSTRUCTIONS

TOKYO OFFICE Ohtemachi Bldg 1-6-1 Ohtemachi, Chiyoda-ku Tokyo, 100 Japan
Phone (03) 3284-9111 Telex YASKAWA J33530 Fax (03) 3284-9034
YASKAWA ELECTRIC AMERICA, INC.
Chicago-Corporate Headquarters 2942 MacArthur Blvd. Northbrook IL 60062-2028 U.S.A.
Phone (708) 291-2340 Fax (708) 498-2430
Chicago-Technical Center 3160 MacArthur Blvd. Northbrook IL 60062-1917, U.S.A.
Phone (708) 291-0411 Fax (708) 291-1018
MOTOMAN INC.
805 Liberty Lane West Carrollton OH 45449, U.S.A.
Phone (513) 847-6200 Fax (513) 847-6277
YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Rua Conde Do Pinhal 8-5° Andar Sala 51 CEP 01501-São Paulo-SP, Brasil
Phone (011) 35-1911 Fax (011) 37-7375
YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone (49) 6196-569-300 Fax (49) 6196-888-301
Motoman Robotics AB
Box 130 S-38500, Torsås, Sweden
Phone 0486-10575 Fax 0486-11410
Motoman Robotec GmbH
Kammerfeldstraße 1 85391 Allershausen, Germany
Phone 08166-900 Fax 08166-9039
YASKAWA ELECTRIC UK LTD.
3 Drum Mains Park Orchardton Woods Cumbernauld Scotland, G68 9LD U.K.
Phone (236)735000 Fax (236)458182
YASKAWA ELECTRIC KOREA CORPORATION
8th Floor Seoul Center Bldg. 91-1 Sogong-Dong, Chung-ku, Seoul Korea 100-070
Phone (02)776-7844 Fax (02)753-2639
YASKAWA ELECTRIC (SINGAPORE) PTE LTD
Head Office : CPF Bldg 79 Robinson Road # 13-05, Singapore 0106, SINGAPORE
Phone 221-7530 Telex (87) 24890 YASKAWA RS Fax 224-5854
Service Center : 221 Henderson Road # 07-20 Henderson Building Singapore 0315, SINGAPORE
Phone 276-7407 Fax 276-7406
YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road Taipei, Taiwan
Phone (02) 563-0010 Fax (02) 567-4677
SHANGHAI OFFICE Room No. 8B Wan Zhong Building 1303 Yan An Road (West), Shanghai 200050 CHINA
Phone (86) 212-1015 Fax (86) 212-1015
TAIPEI OFFICE Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road Taipei Taiwan
Phone (02) 563-0010 Fax (02) 567-4677

YASKAWA ELECTRIC CORPORATION

YASKAWA

TOE-C843-11.20D
© Printed in Japan March 1995 92-2 1WA