



TECHNICAL REPORT

EIA-494™

REV. B

Revised 1992-07
Stabilized 2015-12

32 Bit Binary CL (BCL) and 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines

NOTICE

This document has been taken directly from the original TechAmerica document and contains only minor editorial and format changes required to bring it into conformance with the publishing requirements of SAE Technical Standards. The release of this document is intended to replace the original with the SAE International document. Any numbers established by the original document remain unchanged.

The original document was adopted as an SAE publication under the provisions of the SAE Technical Standards Board (TSB) Rules and Regulations (TSB 001) pertaining to accelerated adoption of specifications and standards. TSB rules provide for (a) the publication of portions of unrevised specifications and standards without consensus voting at the SAE committee level, and (b) the use of the existing specification or standard format.

SAE Technical Standards Board Rules provide that: "This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user."

SAE reviews each technical report at least every five years at which time it may be revised, reaffirmed, stabilized, or cancelled. SAE invites your written comments and suggestions.

Copyright © 2015 SAE International

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

TO PLACE A DOCUMENT ORDER:
Tel: 877-606-7323 (inside USA and Canada)
Tel: +1 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org
<http://www.sae.org>

SAE WEB ADDRESS:

Full document details: [SAE EIA494B](#)

SAE values your input. To provide feedback on this Technical Report, please visit <http://www.sae.org/technical/standards/EIA494B>

Documents

Document Sharing Hub

EIA STANDARD

32 Bit Binary CL (BCL) and 7 Bit ASCII CL (ACL) Exchange Input Format for Numerically Controlled Machines

EIA-494-B
(Revision of EIA-494-A)

JULY 1992

ELECTRONIC INDUSTRIES ASSOCIATION
ENGINEERING DEPARTMENT



NOTICE

EIA Engineering Standards and Publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards and Publications shall not in any respect preclude any member or nonmember of EIA from manufacturing or selling products not conforming to such Standards and Publications, nor shall the existence of such Standards and Publications preclude their voluntary use by those other than EIA members, whether the standard is to be used either domestically or internationally.

Recommended Standards and Publications are adopted by EIA in accordance with the American National Standards Institute (ANSI) patent policy. By such action, EIA does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Recommended Standard or Publication.

This EIA Standard is considered to have International Standardization implication, but the International Electrotechnical Commission activity has not progressed to the point where a valid comparison between the EIA Standard and the IEC document can be made.

This Standard does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this Standard to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

Published by

ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2001 Pennsylvania Ave. N.W.,
Washington, D.C. 20006

**PRICE: Please refer to the current
Catalog of EIA & JEDEC STANDARDS & ENGINEERING PUBLICATIONS
or call Global Engineering Documents, USA and Canada (1-800-854-7179)
International (714-261-1455)**

Copyright 1992

All rights reserved
Printed in U.S.A.

PLEASE!
DON'T VIOLATE
THE
LAW!

This document is copyrighted by the EIA and may not be reproduced without permission.

Organizations may obtain permission to reproduce a limited number of copies through entering into a license agreement with the EIA. For information, contact:

EIA Engineering Publications Office
2001 Pennsylvania Ave., N.W.
Washington, D.C. 20006
(202)457-4963

32 BIT BINARY CL (BCL) AND 7 BIT ASCII CL (ACL) EXCHANGE INPUT FORMAT FOR NUMERICALLY CONTROLLED MACHINES

(From EIA Standard RS-494A and Standards Proposal No. 2918,
formulated under the cognizance of the IE-31 Committee on
Numerical Control Systems and Equipment.)

TABLE OF CONTENTS

FOREWORD	0 - 1
1.0 GENERAL	1 - 1
1.1 PURPOSE	1 - 1
1.2 SCOPE	1 - 1
2.0 BCL COORDINATE SYSTEM	2 - 1
2.1 AXES	2 - 1
2.2 TOOL POSITION	2 - 1
2.3 TOOL AXIS ORIENTATION	2 - 1
2.3.1 Orthogonal Linear Axes	2 - 1
2.3.2 Angularly Positionable Axes	2 - 1
2.3.3 Turning Machine Axes without Live Tools	2 - 3
2.3.4 Turning Machine Axes with Live Tools	2 - 3
2.4 SPINDLE DIRECTION	2 - 6
3.0 CHARACTER DATA	3 - 1
4.0 FILE STRUCTURE	4 - 1
4.1 FILE HEADER	4 - 1
4.2 PARTNO	4 - 1
4.3 UNITS	4 - 1
4.4 RESOLN	4 - 1
4.5 SUBROUTINES	4 - 2
4.6 DATA SEGMENTS	4 - 2
4.7 FINI (END OF FILE) RECORD	4 - 3
5.0 RECORD HEADERS	5 - 1
6.0 RECORD TYPES	6 - 1
6.1 COMMANDS	6 - 1
6.1.1 (8000 nnnn) FINI (End of File) record	6 - 1
6.1.2 (8iii nnnn) General Command record	6 - 1
6.1.3 (8iii nnnn) Text record	6 - 1
6.2 MOTION RECORD TYPES	6 - 2

6.2.1 Motion Records without References to Variables and Registers . . .	6 - 2
6.2.1.1 (90pp nnnn) Destination Point (GOTO) record without Tool Vectors	6 - 2
6.2.1.2 (9100 nnnn) Circular Interpolation (ARC) record	6 - 3
6.2.1.2.1 Major Plane Circular record	6 - 3
6.2.1.2.2 Unrestricted Circular record	6 - 3
6.2.1.3 (92pp nnnn) Destination Point (GOTO) record with Tool Vector(s)	6 - 4
6.2.2 Motion Records with References to Variables and Registers . . .	6 - 4
6.3 UNITS RECORD TYPE	6 - 5
7.0 INTEGER CODES	7 - 1
7.1 COMMAND INTEGER CODES	7 - 1
7.2 MODIFIER INTEGER CODES	7 - 3
8.0 SYNTAX AND SEMANTICS	8 - 1
8.1 SCALE FACTORS	8 - 2
8.2 VARIABLES AND REGISTERS	8 - 2
8.2.1 Origin Table Registers	8 - 4
8.2.2 Probe Registers	8 - 4
8.2.3 Transform Matrix Registers	8 - 6
8.2.4 Tool Data Table Registers	8 - 6
8.3 RECORD DEFINITIONS	8 - 9
9.0 TURNING MACHINE APPLICATIONS	9 - 1
9.1 COORDINATE SYSTEM CONVENTION FOR TURNING	9 - 1
9.2 COORDINATE SYSTEM CONVENTION FOR LIVE TOOLS	9 - 1
9.3 TURNING MACHINE CONFIGURATIONS	9 - 3
9.4 MULTIPLE TOOL SLIDE TURNING MACHINES	9 - 6
9.5 EXAMPLES OF BCL PART PROGRAMS FOR TURNING APPLICATIONS	9 - 6
9.5.1 Dual Tool Slide Turning Machine	9 - 6
9.5.2 Threading Examples	9 - 8
9.5.2.1 Fixed Lead Threading	9 - 8
9.5.2.2 Variable Lead Threading	9 - 10
9.5.2.3 Multi-start Threading	9 - 12
9.5.2.4 Automatic Threading Cycle	9 - 15
9.5.3 Bar Feeding Mechanism	9 - 22
9.6 APPLICATION OF LIVE TOOLS	9 - 22
10.0 BCL FUNCTION CLASSIFICATION	10 - 1
10.1 COMMON FUNCTION SET	10 - 1
10.2 MACHINING CENTER FUNCTION SET	10 - 3
10.3 ROTARY TABLE/LIVE TOOL FUNCTION SET	10 - 5
10.4 MULTI-AXIS PROFILER FUNCTION SET	10 - 6
10.5 EXTENDED CLRSRF FUNCTION SET	10 - 6
10.6 CONSTANT SURFACE SPEED FUNCTION SET	10 - 6
10.7 THREADING FUNCTION SET	10 - 7
10.8 VARIABLE LEAD THREAD FUNCTION SET	10 - 7
10.9 MULTIPLE ENTRY THREAD FUNCTION SET	10 - 7

10.10	EXTENDED CUTCOM FUNCTION SET	10 - 7
10.11	TURNING MACHINE FUNCTION SET	10 - 7
10.12	EXTENDED CIRCULAR INTERPOLATION FUNCTION SET	10 - 8
10.13	COLLINEAR AXES MACHINE FUNCTION SET	10 - 8
10.14	AUTOMATIC THREADING CYCLE FUNCTION SET	10 - 8
10.15	TURNING MACHINE TOOL RETRACT FUNCTION SET	10 - 9
10.16	MULTIPLE TOOL SLIDE MACHINE FUNCTION SET	10 - 9
10.17	CYCLE FUNCTION SET	10 - 9
10.18	BLEND MODE FUNCTION SET	10 - 10
10.19	CLEAR ZONE AND FORBIDDEN ZONE FUNCTION SET	10 - 10
10.20	DATA LOGGING FUNCTION SET	10 - 10
10.21	FLAME CUTTER FUNCTION SET	10 - 11
10.22	BASIC MATHEMATICAL OPERATIONS FUNCTION SET	10 - 11
10.23	EXTENDED MATHEMATICAL OPERATIONS FUNCTION SET	10 - 12
10.24	PROGRAM FLOW CONTROL FUNCTION SET	10 - 12
10.25	SUBROUTINE FUNCTION SET	10 - 12
10.26	BASIC PROBE FUNCTION SET	10 - 12
10.27	EXTENDED (5 AXIS) PROBE FUNCTION SET	10 - 13
10.28	INSERT FUNCTION SET	10 - 13
10.29	MCHTOL FUNCTION SET	10 - 13
11.0	ROTREF EXAMPLES	11 - 1
11.1	SAMPLE BCL/ACL PART PROGRAM USING THE ROTATE/ COMMANDS	11 - 1
11.2	ROTREF EXAMPLES WITH A ROTARY B TABLE	11 - 5
12.0	BCL FORMAT 32 BIT WORD NUMERICAL RANGES	12 - 1
13.0	BYTE SERIAL FILE TRANSFER	13 - 1
14.0	ASCII REPRESENTATION OF BCL (ACL)	14 - 1
14.1	File Structure	14 - 1
14.1.1	Text Line	14 - 1
14.1.2	Line Terminator	14 - 1
14.2	File Header	14 - 1
14.2.1	Initial Header Line	14 - 1
14.2.2	Optional Header Lines	14 - 1
14.3	ACL Record	14 - 2
14.3.1	Record Initial Text Line	14 - 2
14.3.2	Record Continuation Text Line	14 - 2
14.3.3	Optional	14 - 2
14.3.4	Reference Number Field	14 - 2
14.3.5	Record Command Fields	14 - 2
14.3.5.1	BCL Commands	14 - 2
14.3.5.2	Multax Command	14 - 2
14.3.6	Record Parameter Field	14 - 3
14.3.6.1	Modifier Field	14 - 3
14.3.6.2	Numerical Field	14 - 3
14.3.6.3	Text Field	14 - 3

14.3.7 Parameter Separator Field	14 - 3
14.4 Whitespace Characters	14 - 4
14.5 Text Line continuation	14 - 4
14.5.1 Continuation Character	14 - 4
14.6 Case Sensitivity	14 - 4
APPENDIX A - ASCII 7 BIT CODE SET (ANSI X3.4)	Appendix A - 1
APPENDIX B - GLOSSARY OF TERMS	Appendix B - 1
APPENDIX C - REQUESTS FOR EXTENSIONS TO THE STANDARD	Appendix C - 1
APPENDIX D - SAMPLE BCL PROGRAM	Appendix D - 1
APPENDIX E - RELATED STANDARDS AND STANDARDS PROPOSALS	Appendix E - 1
COMMAND WORD INDEX	Command word index - i
MODIFIER WORD INDEX	Modifier word index - i

STANDARDS FOR 32 BIT BINARY CL (BCL) AND 7 BIT ASCII CL (ACL) EXCHANGE INPUT FORMAT FOR NUMERICALLY CONTROLLED MACHINES

FOREWORD

The information and comments contained in this foreword should not be considered part of this standard. They have been included for clarification and guidance.

This revision of EIA-494 incorporates the addition of the ASCII or text representation of BCL, which is termed ACL. It has been added to provide for a standard human-readable representation and to permit the ready transfer of BCL part program files among systems that have limited or awkward binary input/output capability (e.g. most mainframes and some minicomputers). In addition, the standardization of the text form of BCL (ACL) permits the conversion of BCL into text editable form and the ready conversion of the edited text back into BCL without degradation of data.

BINARY CL EXCHANGE (BCL) FORMAT input for numerical control provides a standardized part oriented format that allows different machines to operate from the same input data.

The fundamental principle guiding this standard is that the orientation of the input data is toward the part coordinate system.

Use of BCL FORMAT requires that the numerical control machining system perform all machine geometry dependent functions (which may have previously been performed by a postprocessor).

BCL format represents NC machining input data as a series of records that are groups of 32 bit binary integer words.

A BCL file may be produced by a variety of methods. Three methods are by converting a CLfile to a BCL file from an APT or CAD system, outputting a BCL file directly from a CAD system; or by de-posting an RS-274 part program. A statement such as "MACHIN/BCL" may be required to invoke a BCL converter on an APT system.

The part oriented postprocessing program, which converts the contents of a CLfile to the BCL FORMAT file, is called a BCL CONVERTER. Its primary functions are: to retrieve CLDATA records, select the relevant portions of the data, convert the floating point data to 32 bit integer data, format data into records as described in Section 6.0 (RECORD TYPES), produce a BCL program file and a program listing. It performs no machine-dependent functions.

The BCL format was originally developed for use with the APT part program processor. It is designed to utilize the integer codes defined for major and minor words from the ANSI X3.37 specified CLDATA. All postprocessor commands are represented as integer codes. Major word integer codes are carried in the header, minor word integer codes follow the header and may be mixed with numerical and character data.

This standard covers the format of the BCL input data for numerically controlled machines and is not intended to control the logical format of APT source statements nor to specify machine design.

Data exchangeability between machines requires that certain provisions be considered. While this standard provides a BCL data format intended to foster exchangeability, some of the other considerations are:

- A. The machine tools should be of like configuration, but not necessarily of the same model or built by the same manufacturer.
- B. The machine should have the requisite size, rigidity, accuracy, spindle horsepower, speed, etc., to produce the parts to which the data is intended to apply.
- C. Fixtures required to hold the part in the proper manner should be available for the alternative machine or for transfer with the part assignment to the alternative machine.
- D. The clamping arrangement and location of clamps should be the same for the principal and alternative machines when making the part in question.
- E. Tooling for the alternative machine should be available and suitable for production of the intended part.
- F. The numerical control system must be capable of responding correctly to BCL data format and commands as specified in this standard and as they apply to the particular machine type. It is the implementor's responsibility to provide an appropriate error action in response to illegal or unrecognized commands.
- G. If a portable input medium is used for input of BCL data, the medium format must be considered.
- H. If a data link is used for input of BCL data, the electrical characteristics, line control protocol, and user language must be considered.
- I. The work envelope should be as small as possible. All clearances should be as close to the work as possible.
- J. The part/tool should always be programmed, not the machine. The selection of feeds and speeds should be based on part and finish.
- K. The part program zero is always related to the part coordinate system. The ORIGIN/ command will relate the part program zero to the machine zero.
- L. Tool data and machine specific data is contained in BCL files (external subroutines) separate from the machine independent part program files.
- M. Adherence to flexible disk cartridge standards ISO/DIS 9293 for volume and file structure, ISO/DIS 8860/1 for dimensional, physical, and magnetic characteristics, and ISO/DIS 8860/2 for track format, permits maximum exchangeability between various computer systems. (ISO documents may be obtained from:

Global Engineering Documents
2625 Hickory St.
Santa Ana, CA 92707
714-540-9870
800-854-7179

or

American National Standards Institute
1430 Broadway
New York, NY 10018

1.0 GENERAL

1.1 PURPOSE

The purpose of this Standard is to define a uniform, part oriented data format for the distribution of machine input data to numerically controlled machines, and the response of the machines thereto, in order to permit the exchange of such input data among a wider variety of machines than is possible with existing Standards.

1.2 SCOPE

The scope of this Standard is the definition of the response of a numerically controlled machine to a valid sequence of records made up of 32 bit binary words or ASCII text strings. The Standard defines the structure of these records and of the 32 bit binary words or ASCII text strings which make up the records.

This standard addresses the control of machines capable of performing 2, 3, 4, and 5 axis motion of an active tool (mill, laser, pen, etc.) relative to a part, and those capable of 2 and 4 axis tool motion relative to a rotating part (turning machines), including parallel tool slide sets capable of concurrent (merged) motion.

2.0 BCL COORDINATE SYSTEM

The BCL coordinate system defines the position and orientation of a tool relative to a part.

All references in this Standard to machine coordinates conform to ANSI/EIA RS-267-C AXIS AND MOTION NOMENCLATURE FOR NUMERICALLY CONTROLLED MACHINES.

2.1 AXES

The X, Y, and Z axes of the BCL coordinate system, as applied to a part, form a right hand coordinate system as shown in figures 2-1 and 2-2.

2.2 TOOL POSITION

The position of the tool with respect to the part is described in BCL coordinates X, Y, and Z as applicable.

2.3 TOOL AXIS ORIENTATION

2.3.1 Orthogonal Linear Axes

Orientation of the tool axis in a 2 or 3 axis system of orthogonal linear axes using an active tool (e.g. mill, drill, laser beam, etc.) is parallel to the Z axis of the BCL coordinate system.

2.3.2 Angularly Positionable Axes

Orientation of the tool axis in a system with additional angularly positionable axes (e.g. rotary table, gimbaled spindle, servo controlled turret, right angle attachment, etc.) is defined by the direction cosines of the tool axis vector.

The tool axis vector is a unit vector pointing in a direction from the tool tip back along the tool and colinear with the center line of the tool (figure 2-1).

The direction cosines (i , j , k) are the signed values of the projections of the tool axis unit vector on the X, Y, and Z axes respectively of the BCL coordinate system.

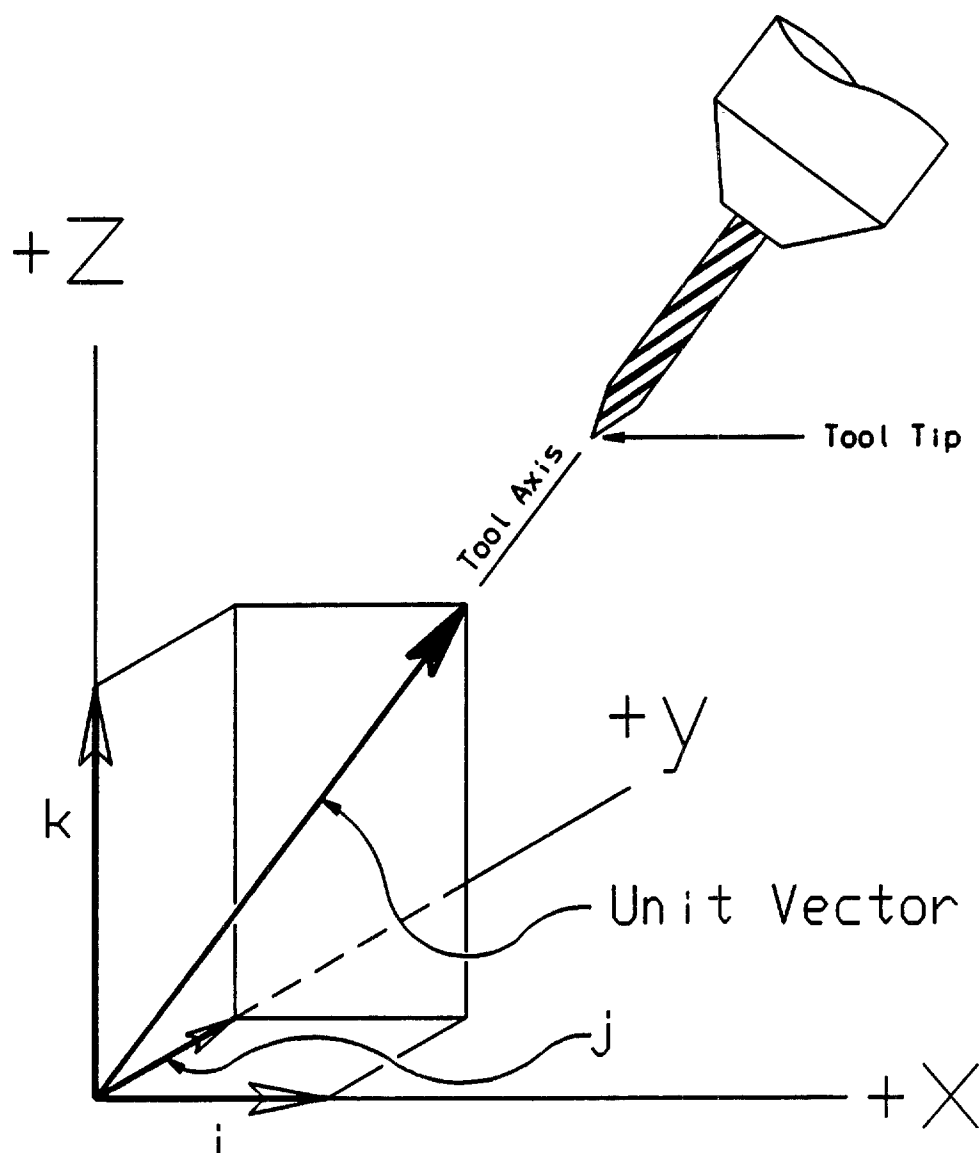


FIGURE 2-1

DIRECTION COSINES OF A TOOL AXIS VECTOR

The direction cosines are the components of the unit vector parallel to the tool axis:

- i is the component parallel to the x axis,
- j is the component parallel to the y axis,
- k is the component parallel to the z axis.

2.3.3 Turning Machine Axes without Live Tools

Orientation of the axis of rotation for a turning machine is along the Z axis of the BCL coordinate system. Radial motion of the tool is along the X axis of the BCL coordinate system. The Y value associated with a point in a BCL record shall be zero.

2.3.4 Turning Machine Axes with Live Tools

Live or rotating tools for turning machines are used to perform milling on a turning machine. Figures 2-2 and 2-3 define the BCL coordinate system for a turning machine with live tools. Note that the coordinate system is not dependent upon whether the tool is oriented along the X or Z axis of the machine. Refer to Section 2.3.2 for a method of controlling the orientation of the tool axis.

If the only orientation of the tool axis is parallel to the X or Z machine axis (as specified by the XAXIS or ZAXIS modifiers, respectively, in the tool data table), then the BCL part program shall not contain tool axis vectors.

The C axis is rotated so that the tool is positioned at the commanded x and y values. The machine x and c values are calculated per the equations that follow:

$$X_{machine} = \sqrt{X^2 + Y^2} \quad C_{machine} = \arctan \left(\frac{X}{Y} \right)$$

For the program which requires the capability of rotating or indexing the tool to other than the +Z machine axis orientation, tool axis vectors are required. If the angle B is the amount of rotation of the tool from the +Z machine axis, then the 'k' component of the tool axis vector is given by:

$$k = \cos(B)$$

A positive 90 degrees for B results in the tool aligned with the +X machine axis.

The 'i' and 'j' components of the tool axis vector are calculated from the X and Y components of destination point records as follows:

$$i = \frac{X \sqrt{1 - k^2}}{\sqrt{X^2 + Y^2}} \quad j = \frac{Y \sqrt{1 - k^2}}{\sqrt{X^2 + Y^2}}$$

where 'i' has the same sign as X and 'j' has the same sign as Y.

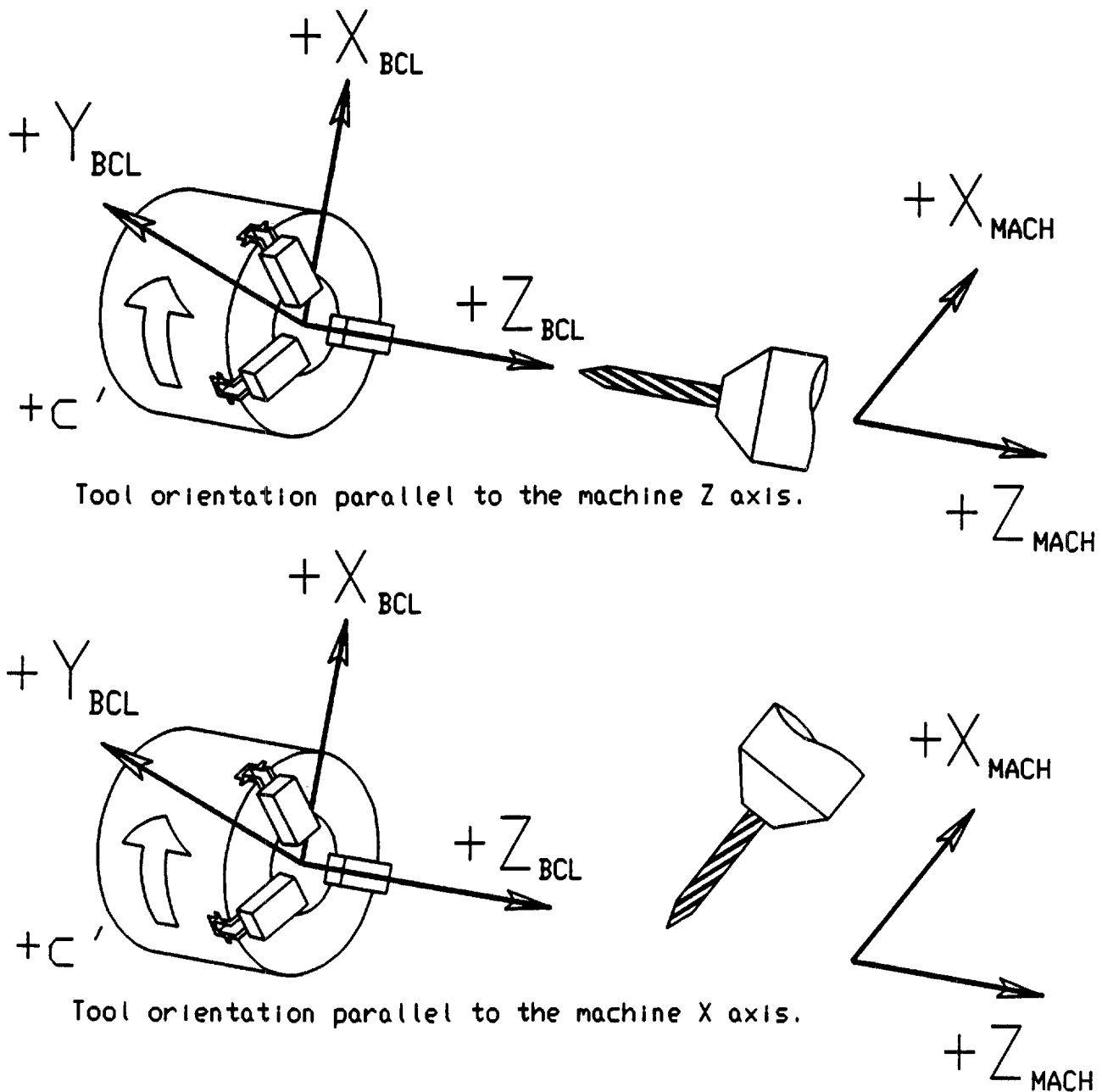


FIGURE 2-2

BCL COORDINATE SYSTEM FOR TURNING MACHINES WITH LIVE TOOLING

The BCL Z axis is always aligned with the machine Z axis.
 The coordinate system is not affected by tooling type or
 the orientation of live tooling.
 Rotary motion (C' axis) is used to reach non-zero Y positions.

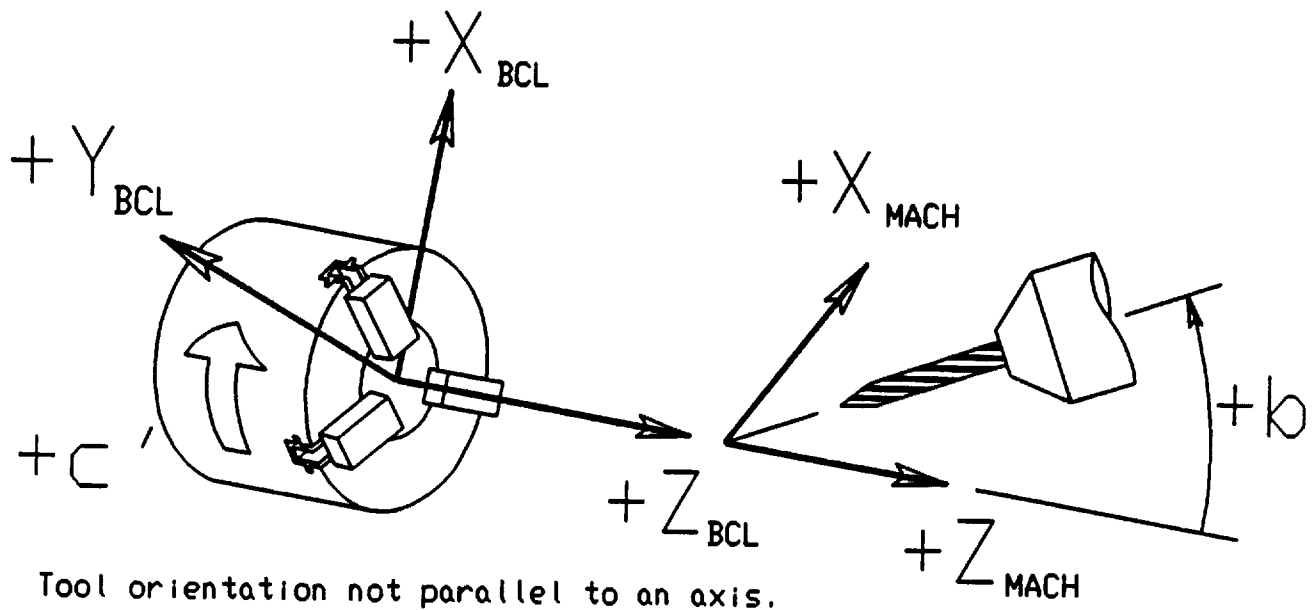
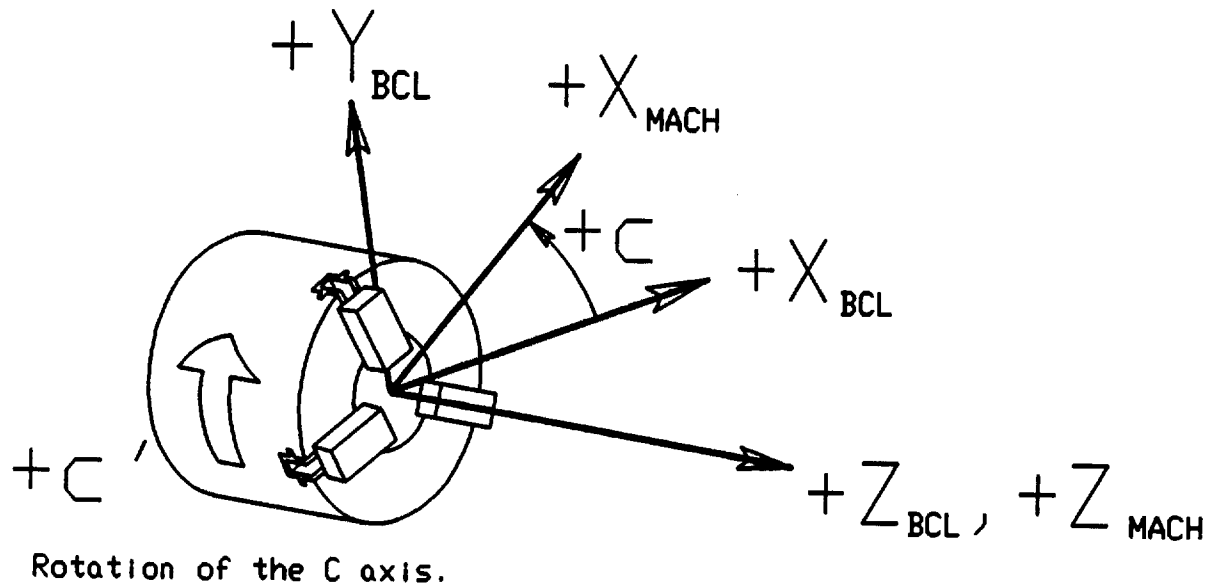


FIGURE 2-3

ANGULAR MEASUREMENTS WITH LIVE TOOLING

The C' axis rotation angle is zero when the BCL X axis and the machine X axis are aligned. The tool axis angle is measured in the ZX plane, with zero along the Z axis.

2.4 SPINDLE DIRECTION

The clockwise (CLW) direction of the spindle shall be as defined in ANSI/EIA RS-267-C: a clockwise spindle rotation is in the direction which advances a right-handed screw into the workpiece. In the case of a turning machine, this is the direction of a right-handed screw, located in the headstock, as it is advanced into a right-handed nut. Turning machines may use NRMTRN (normal turn) and REVTRN (reverse turn) as direction indicators.

3.0 CHARACTER DATA

BCL character data shall consist of one or more 32 bit words. Each word shall consist of 4 characters stored left to right (i.e. the first character in the most significant position, the fourth character in the least significant position). Each character shall be 8 bits in length with the ASCII 7 BIT CODE (ANSI X3.4) in the 7 least significant bits and the high order bit set to zero (see Appendix A).

A character in the range of 40 hex through 5F hex preceded by a ~ (7E hex) shall have 40 hex subtracted from it (to become a control character), except ~~ is interpreted as a ~.

The 32 bit word shall be padded with trailing space characters as necessary to fill a 32 bit word.

See Section 13.0 for bit sequence requirements for BCL file transfer procedures.

4.0 FILE STRUCTURE

Each input file in BCL format comprises the following elements in the order shown:

```

FILE HEADER
PARTNO
UNITS
RESOLN/n          (optional)
SUBROUTINE(S)     (optional)
DATA SEGMENT(S)
FINI (END OF FILE)

```

4.1 FILE HEADER

The file header shall be composed of BCL character data as defined in Section 3.0. The file header may be any integer multiple of 4 characters greater than or equal to 12 characters. The first 1024 characters are reserved for descriptive data associated with the BCL file.

The fields of data are defined as follows:

<u>REQUIRED HEADER FIELDS</u>	<u>POSITION</u>	<u>CONTENTS</u>
BCL file identifier	1-6	EIA494
SP character	7	SP
Revision Level	8	A thru Z
SP character	9	SP
File Type	10-12	PRI or AUX
	13-1024	Reserved for future use and, if present, filled with SP characters.

Note: SP is a space character.

A BCL file is identified by the characters EIA494 as the first six characters of the file header. The Revision Level field specifies conformance of the file to a revision level of this standard. The file type specifies either a Primary (PRI) or Auxiliary (AUX) command file (refer to Section 8.3).

4.2 PARTNO

A PARTNO record shall appear immediately following the file header.

4.3 UNITS

A UNITS record shall appear immediately following the PARTNO record.

4.4 RESOLN

The optional RESOLN/n record specifies a modal scaling if different from the default inch or metric scaling set by the UNITS record.

4.5 SUBROUTINES

The optional embedded subroutines are delineated by DEFSUB-ENDSUB pairs.

4.6 DATA SEGMENTS

Each BCL data segment consists of one or more BCL data records. Only machines with more than one independent tool slide may have more than one data segment, and then no more than one per independent tool slide.

Normally there is only one independent tool slide, even in the case of an ID and OD turret mounted on a common slide. A 4 axis turning center is an example of two independent turrets, and shall have a data segment associated with each turret programmed.

For the case of more than one data segment, the first record of each data segment shall be a SELECT/TLIDLID,n command. The termination of all data segments is specified by an END command.

For the case of only one data segment, a SELECT/TLIDLID,n command is not required.

Each BCL data record consists of one or more 32 bit binary words representing commands, modifiers, and associated data. The BCL record number is also included in each record (see Section 6.1.2 for the format of commands and modifiers).

Each BCL record begins with a 32 bit record header representing the record function and record number. The two high order bits of BCL words are reserved for header identification (see Section 12.0 for numerical ranges of BCL words), thus headers are distinguished from other 32 bit words in the file. A header thus delimits the preceding record.

The 32 bit words following the header, if any, are interpreted as required by the function specified by the first word. These words may be modifiers, numerical data, or character data. The format for each record type is given in the following sections.

Numerical data is represented as a 32 bit binary integer. The 31 low order bits are utilized for numerical data in the range of -2 raised to the 30th power to $+2$ raised to the 30th power.

Numerical data other than vector components and scale factors is scaled by a default factor of 10,000 if system units are INCHES, and 1,000 if system units are MM. The scale factor may be altered by the RESOLN/n record or otherwise specified by SCALE modifiers within certain records.

Scale factors are scaled by a factor of 1.

Vector components are scaled by a factor of 2 raised to the 30th power. For example, a vector component value of 1 would be expressed as an integer value of 1,073,741,824 (40 00 00 00 HEX).

Negative numerical data is expressed as the two's complement (of all 32 bits) of the positive value.

Text records contain an integer multiple of four characters (32 bits) in order to avoid partial words (less than 32 bits). Incomplete words are padded with trailing space characters (see Section 3.0).

4.7 FINI (END OF FILE) RECORD

The input file ends with a FINI (END OF FILE) record.

5.0 RECORD HEADERS

The first word of each data record is a header consisting of a 32 bit binary word which defines the record function and the record number. The format for each is given below. The numerical values between parentheses are shown in hexadecimal notation.

```
(8000 nnnn) End of File (FINI) record
              (nnnn) = Record Number

(8iii nnnn) Command record
              ( iii) = Command Integer Code
              (nnnn) = Record Number

(90pp nnnn) Destination Point (GOTO) record
              without Tool Vectors
              (  pp) = Number of Points
              (nnnn) = Record Number

(9100 nnnn) Circular Interpolation (ARC) record
              (nnnn) = Record Number

(9101 nnnn)
  through   - Reserved for future record header types
(91FF nnnn)

(92pp nnnn) Destination Point (GOTO) record
              with Tool Vectors
              (  pp) = Number of Points
              (nnnn) = Record Number

(9300 nnnn)
  through   - Reserved for future record header types
(9408 nnnn)

(9409 nnnn) UNITS record
              (nnnn) = Record Number

(940A nnnn)
  through   - Reserved for future record header types
(9FFF nnnn)
```

6.0 RECORD TYPES

Record types for BCL format files are defined as follows. The numerical values between parentheses are in hexadecimal notation.

6.1 COMMANDS

- 6.1.1 (8000 nnnn) FINI (End of File) record
(nnnn) = Record Number
- 6.1.2 (8iii nnnn) General Command record
(iii) = Command Integer Code (see sec. 7.1)
(nnnn) = Record Number

A general command record header may be followed by zero or more 32 bit parameters in the following format:

(dddd dddd) or (7FFF mmmm)

(dddd dddd) Numerical Data
(7FFF mmmm) Modifier
(7FFF) = Modifier Identifier
(mmm) = Modifier Integer Code
(see Section 7.2)

- 6.1.3 (8iii nnnn) Text record
(iii) = command
- (3FD) = DISPLY
(414) = PPRINT
(415) = PARTNO
(416) = INSERT
- (nnnn) = Record Number

A text record header may be followed by zero or more 32 bit words, each containing four characters of BCL character data. The last 32 bit word shall contain trailing space characters, as necessary, to complete the 32 bits.

6.2 MOTION RECORD TYPES

The following records define linear and circular motions of the tool in the BCL coordinate system.

All coordinates in the following records are absolute dimensions in the BCL coordinate system. No incremental destination point records are provided.

All destination point records are executed at the feedrate specified by the FEDRAT/ command or RAPID command, whichever is in effect, except during CYCLES (see CYCLE/ in Section 8.3).

All circular records are executed at the feedrate specified by the FEDRAT/ command.

6.2.1 Motion Records without References to Variables and Registers

6.2.1.1 (90pp nnnn) Destination Point (GOTO) record without Tool Vectors

(pp) = Number of Points
(nnnn) = Record Number

```

(xxxx xxxx) x coordinate }
(yyyy yyyy) y coordinate } Point No. 1
(zzzz zzzz) z coordinate }
      .           .           .
      .           .           .
      .           .           .
      .           .           .
(xxxx xxxx) x coordinate }
(yyyy yyyy) y coordinate } Point No. pp
(zzzz zzzz) z coordinate }
```

This record defines (in the BCL coordinate system) the end point of a straight line from the previous tool location along which the tool motion is linearly interpolated.

The default or initial tool axis orientation at the beginning of a BCL program is 0,0,1.

A tool axis vector established by a previous (92pp nnnn) destination point record is modal and all coordinates shall be transformed as if each (90pp nnnn) record contained the modal tool vector.

6.2.1.2 (9100 nnnn) Circular Interpolation (ARC) record (nnnn) = Record Number

```

(xxxx xxxx)      x coordinate
(yyyy yyyy)      y coordinate   } center of arc
(zzzz zzzz)      z coordinate

(iiii iiii)      i coordinate   } vector perpendicular
(jjjj jjjj)      j coordinate   } to plane of
(kkkk kkkk)      k coordinate   } interpolation

(xxxx xxxx)      x coordinate   } end point of arc
(yyyy yyyy)      y coordinate   }
(zzzz zzzz)      z coordinate   }
```

This record defines the center, plane and direction, and end point of a circular arc tool path (the start point of the arc is defined by the preceding end point of another line or arc). The control shall use circular interpolation to define the arc.

6.2.1.2.1 Major Plane Circular record

If the end point is the same as the start point a full 360 degree circle is generated.

Circular interpolation is permitted only in a plane which is parallel to one of the XY, YZ, or ZX planes of the BCL coordinate system.

The i, j, and k values define the plane and direction of motion as follows:

i	j	k	PLANE	DIRECTION	VIEWED FROM
0	0	1	XY	CCLW	+Z
0	0	-1	XY	CLW	+Z
1	0	0	YZ	CCLW	+X
-1	0	0	YZ	CLW	+X
0	1	0	ZX	CCLW	+Y
0	-1	0	ZX	CLW	+Y

6.2.1.2.2 Unrestricted Circular record

The plane of circular interpolation is perpendicular to a direction vector defined by the i, j, and k values. The direction of interpolation shall be clockwise when viewed in the direction of the vector.

If the end point of the preceding record and the end point of the circular record do not lie in this same plane, the interpolated path shall describe a helix about a line through the center point (defined by the circular record) and parallel to the direction vector. The helix shall terminate at the end point defined by the circular record.

6.2.1.3 (92pp nnnn) Destination Point (GOTO) record with Tool Vector(s)
 (pp) = Number of Point
 (nnnn) = Record Number

(xxxx xxxx)	x	coordinate	} Point No. 1
(yyyy yyyy)	y	coordinate	
(zzzz zzzz)	z	coordinate	
(iiii iiii)	i	direction cosine	
(jjjj jjjj)	j	direction cosine	
(kkkk kkkk)	k	direction cosine	
.	.	.	.
.	.	.	.
.	.	.	.
(xxxx xxxx)	x	coordinate	} Point No. pp
(yyyy yyyy)	y	coordinate	
(zzzz zzzz)	z	coordinate	
(iiii iiii)	i	direction cosine	
(jjjj jjjj)	j	direction cosine	
(kkkk kkkk)	k	direction cosine	

This record defines the end point and ending direction cosines of the tool tip and tool axis when controlling a machine with one or two rotary axes.

The interpolation of the path from the preceding destination point and direction cosines shall be linear in part space if linearization (LINTOL/) is on and linear in machine space if linearization is off (see LINTOL/ in Section 8.3).

This record establishes a modal tool axis vector for use by subsequent destination point (GOTO/, MOVTO/) records without tool axis vectors.

6.2.2 Motion Records with References to Variables and Registers

Single destination point and major plane circular motion records that utilize variables and probe registers as modifiers are defined by MOVTO/ and MOVARC/ commands in Section 8.3. The resultant motion is per the definition in Section 6.2.1.

6.3 UNITS RECORD TYPE

This record sets the system units and the default scale factor (see Appendix D line D.2 for example).

(9409 nnnn) UNITS record
(nnnn) = Record Number

A UNITS record header is followed by one modifier, either INCHES(12F) or MM(12D), in the following format:

(7FFF mmmmm) Modifier
(7FFF) = Modifier Identifier
(mmmmm) = Modifier Integer Code

The global scaling of parameters other than direction cosines is 10,000 for inch and 1,000 for mm. For values different from the default for [INCHES] or [MM] a RESOLN/n record shall immediately follow the UNITS/ record.

The scale for direction cosines shall be 2^{30} .

7.0 INTEGER CODES

7.1 COMMAND INTEGER CODES

COMMAND	INTEGER CODE		MEANING
	DECIMAL	HEX	
AIR	1011	3F3	AIR
BARFED	3001	BB9	BAR FEED
BRANCH	3002	BBA	BRANCH
CALC	3003	BBB	CALCULATE
CALSUB	1142	476	CALL SUBROUTINE
CATCHR	3004	BBC	PARTS CATCHER
CHECK	1023	3FF	CHECK DEVICE, PART, OR PROBE
CLAMP	1074	432	CLAMP AXIS
CLRSRF	1057	421	CLEARANCE PLANE
CLRZON	3128	C38	CLEAR ZONE
COMPAR	3005	BBD	COMPARE
COOLNT	1030	406	COOLANT
CUTCOM	1007	3EF	CUTTER RADIUS COMPENSATION
CUTOFF	3006	BBE	CUTOFF SLIDE
CYCLE	1054	41E	CYCLE
DEFSUB	1140	474	DEFINE SUBROUTINE
DELAY	1010	3F2	DELAY
DISPLY	1021	3FD	DISPLAY
ELSE	3007	BBF	ELSE
END	1	1	END
ENDIF	3008	BC0	END IF
ENDLP	3009	BC1	END LOOP
ENDSUB	1141	475	END SUBROUTINE
FEDRAT	1009	3F1	FEEDRATE
FLAME	3010	BC2	FLAME MACHINE
GOHOME	17	11	GO HOME
HOMEPT	3011	BC3	HOME POINT
INSERT	1046	416	INSERT
LABEL	3012	BC4	LABEL
LINTOL	1067	42B	LINEARIZATION TOLERANCE
LOAD	1075	433	LOAD TOOL
LOADPT	3104	C20	LOAD TOOL POSITION
LOOP	3013	BC5	LOOP
MCHTOL	1016	3F8	MACHINE TOLERANCE
MODE	1003	3EB	MODE
MOVARC	3014	BC6	MOVE ALONG ARC
MOVTO	3015	BC7	MOVE TO POINT
NOSE	3016	BC8	SPINDLE NOSE
OPSKIP	1012	3F4	OPTIONAL SKIP
OPSTOP	3	3	OPTIONAL STOP
ORIGIN	1027	403	ORIGIN
PARTNO	1045	415	PART PROGRAM IDENTIFIER
PARTOP	3018	BCA	PART OPERATION
POSTN	1024	400	POSITION SET
PPRINT	1044	414	PROGRAM PRINT

COMMAND INTEGER CODES CONTINUED

COMMAND	INTEGER CODE		MEANING
	DECIMAL	HEX	
PROGLG	3020	BCC	PROGRAM LOG
RAPID	5	5	RAPID
RESOLN	3138	C42	RESOLUTION
RETRCT	7	7	RETRACT
RETURN	3021	BCD	RETURN
ROTATE	1066	42A	ROTATE
SELECT	1101	44D	SELECT
SEQNO	1019	3FB	SEQUENCE NUMBER
SET	1087	43F	SET
SPINDL	1031	407	SPINDLE
STDYRS	3022	BCE	STEADY REST
STOP	2	2	STOP
SYNCTR	3023	BCF	SYNCHRONIZE TOOL SLIDES
TESTIF	3024	BD0	TEST IF
THREAD	1036	40C	THREAD
TLRTCT	3124	C34	TOOL RETRACT
TLSTCK	3025	BD1	TAIL STOCK
TOOLNO	1025	401	TOOL NUMBER
TURRET	1033	409	TURRET

7.2 MODIFIER INTEGER CODES

MODIFIER	INTEGER CODE	
	DECIMAL	HEX
AAXIS	140	8C
ABSENT	3026	BD2
ADJUST	159	9F
AHIT	3029	BD5
ALL	51	33
ANGLE	236	EC
ANOM	3143	C47
ASSIGN	3030	BD6
ATANGL	1	1
AUTO	88	58
BAXIS	141	8D
BHIT	3033	BD9
BLEND	3040	BE0
BNOM	3144	C48
BORE	82	52
BOTH	83	53
CAXIS	142	8E
CCLW	59	3B
CGOTO	3137	C41
CHIT	3036	BDC
CHUCK	1073	431
CLEANP	3038	BDE
CLEARV	205	CD
CLOSE	3039	BDF
CLW	60	3C
CNOM	3145	C49
CONSTP	3041	BE1
CROSS	3	3
CSINK	256	100
CSS	319	13F
DECR	65	41
DEEP	153	99
DELTAX	3042	BE2
DELTAZ	3043	BE3
DEPTHV	201	C9
DOT	3044	BE4
DPM	3045	BE5
DRILL	163	A3
ENTANG	3046	BE6
EQUALP	3047	BE7
FACE	81	51
FEED	248	F8
FINDEP	3048	BE8
FLAG	3049	BE9
FLOOD	89	59
FPM	322	142
FPR	323	143
FPT	3050	BEA
FSTDEP	3051	BEB

MODIFIER INTEGER CODES CONTINUED

MODIFIER	INTEGER CODE	
	DECIMAL	HEX
GAGE	324	144
HEAD	1002	3EA
HIGH	62	3E
HOLDER	157	9D
IDSTRG	3136	C40
IN	48	30
INCHES	303	12F
INCR	66	42
IPM	73	49
IPR	74	4A
IPT	3052	BEC
LARGE	7	7
LEAD	326	146
LEFT	8	8
LENGTH	9	9
LIVE	3034	BDA
LOCK	114	72
LOW	63	3F
MANUAL	158	9E
MASTER	3053	BED
MAXRPM	79	4F
MIST	90	5A
MM	301	12D
MMPM	315	13B
MMPR	316	13C
MMPT	3055	BEF
MULTRD	119	77
NODRAG	289	121
NOMORE	53	35
NONE	236	EC
NRMTRN	3056	BF0
NULINE	3000	BB8
NUPASS	3057	BF1
OFF	72	48
ON	71	47
OPEN	50	32
ORG	3059	BF3
ORGADJ	3027	BD3
ORIENT	246	F6
OUT	49	31
PABS	3063	BF7
PACOS	3066	BFA
PADD	3064	BF8
PAND	3065	BF9
PART	166	A6
PASIN	3067	BFB
PATAN	3068	BFC
PATAN2	3069	BFD
PBSTAT	3100	C1C

MODIFIER INTEGER CODES CONTINUED

MODIFIER	INTEGER CODE	
	DECIMAL	HEX
PCOS	3070	BFE
PDIV	3071	BFF
PEQ	3072	C00
PERPTO	18	12
PEXP	3073	C01
PEXPE	3074	C02
PGE	3075	C03
PGT	3076	C04
PHIT	3077	C05
PILOTL	3078	C06
PINTGR	3079	C07
PLE	3080	C08
PLOG	3081	C09
PLOGE	3082	C0A
PLT	3083	C0B
PMOD	3084	C0C
PMULT	3085	C0D
PNE	3086	C0E
PNEG	3087	C0F
PNOM	3088	C10
PNOT	3089	C11
POR	3090	C12
PPOWER	3091	C13
PRBSET	3092	C14
PROBE	3093	C15
PROUND	3097	C19
PSIN	3098	C1A
PSQRT	3099	C1B
PSUB	3101	C1D
PTAN	3102	C1E
PULBOR	218	DA
PULFAC	223	DF
PXOR	3120	C30
QUILL	3142	C46
RADIUS	23	17
RANDOM	174	AE
RANGE	145	91
REAM	262	106
REV	97	61
REVTRN	3096	C18
RIGHT	24	18
ROTREF	68	44
RPM	78	4E
SAME	54	36
SCALE	25	19
SCRIBE	129	81
SFM	115	73
SIZE	196	C4
SMALL	26	1A

MODIFIER INTEGER CODES CONTINUED

MODIFIER	INTEGER CODE	
	DECIMAL	HEX
SMM	505	1F9
TABLE	177	B1
TAP	168	A8
TEETH	3103	C1F
THRU	152	98
TIMES	28	1C
TIPLOC	3105	C21
TLANGL	294	126
TLDEAD	3139	C43
TLINDX	3148	C4C
TLLIFE	273	111
TLSLID	3106	C22
TLTIME	3140	C44
TLUSER	3141	C45
TO	69	45
TOOL	3017	BC9
TORCH	3107	C23
TOTDEV	3108	C24
TRAV	154	9A
TRIP	3109	C25
TURN	80	50
TYPE	98	62
UAXIS	227	E3
UNIT	30	1E
UNTRIP	3110	C26
VARBL	3111	C27
VARBLP	3112	C28
VAXIS	228	E4
WAXIS	229	E5
WTDANG	3113	C29
XAXIS	84	54
XFORM	3116	C2C
XFRM	3019	BCB
XHIT	3117	C2D
XLENTH	3118	C2E
XNOM	3119	C2F
XORG	3060	BF4
XORGA	3028	BD4
XTM	3129	C39
XWEAR	3121	C31
XWRMAX	3122	C32
XYPLAN	33	21
YAXIS	85	55
YHIT	3125	C35
YLENTH	3126	C36
YNOM	3127	C37
YORG	3061	BF5
YORGA	3031	BD7
YTM	3146	C4A

MODIFIER INTEGER CODES CONTINUED

MODIFIER	INTEGER CODE	
	DECIMAL	HEX
YZPLAN	37	25
ZAXIS	86	56
ZHIT	3130	C3A
ZLENTH	3131	C3B
ZNOM	3132	C3C
ZORG	3062	BF6
ZORGA	3032	BD8
ZTM	3147	C4B
ZWEAR	3134	C3E
ZWRMAX	3135	C3F
ZXPLAN	41	29

8.0 SYNTAX AND SEMANTICS

This section defines the syntax and semantics of the BCL records and their Binary (BCL) and ASCII (ACL) representations.

This section does not define the APT processor input but rather uses APT source conventions as a convenient means for defining the language. The intent of this section is to define the response of the BCL input numerical control unit to specific code sequences or records. The records may have been created by processing an APT Source program, by operation of a graphics processor, or any of a number of other methods.

The following conventions are used to define the syntax of the commands:

1. Elements - appear in both ASCII and hex representations

UPPERCASE WORDS = ASCII symbol

(Integer Code) = hex value

ONE UPPERCASE CHARACTER = triple register

one lowercase character = numerical value, variable,
or storage register

- may be preceded by 'd' (e.g. dx, dy, dz) to emphasize usage as an incremental value
- may be followed by a single digit (e.g. x1, x2, x3) where a sequence of related values is required

'string' = character data with a maximum length of 72

BCL label = IDSTRG(C40),n,'labeltext'

IDSTRG is the modifier prefix for a character string.

'n' is the length of the character string.

'labeltext' is a restricted set of character data with a maximum length of 32.

elements are: upper and lower case letters (A-Z, a-z), numbers (0-9), dollar sign (\$), underscore (_), hyphen (-), ampersand (&), and period (.).

ACL label = Text Field (see 14.3.6.3)

2. Punctuation - appears in ASCII representation only

- / separates the major word from the remainder of the command
- , separates subsequent elements
- indicates continuation of the command string from the previous line
- ... indicates that zero or more repetitions of the preceding element or couplet are permitted
- [modifier] indicates that one of the vertically aligned modifiers must be selected
- {modifier} indicates that one of the vertically aligned modifiers may be selected
- <modifier> indicates the default modifier if one is not selected
- @modifier@ indicates that any element of a modifier set may be selected

"Couplets", as used in this section, refers to a modifier followed immediately by a related scalar or modifier.

The order of parameters following the command integer code is optional unless otherwise stated, except couplets must appear with the modifier immediately followed by a related scalar or modifier.

If a command record per Section 10.0 is detected, and the feature is not available on that numerical control unit system, an appropriate error action as defined by the implementor shall be performed by the numerical control unit. An appropriate error action might be to display a message and execute the functions of a program stop. Optionally, the numerical control unit might allow the operator to specify ignoring the feature during program execution.

8.1 SCALE FACTORS

Within the command records defined in Section 8.3, the scale factor in effect for any single numerical value may be replaced by a scale factor 's' by preceding the numerical value with a {SCALE(19),s,} couplet. The {SCALE,s,} couplet shall change the scale factor of the next subsequent numerical value only (for an example see Appendix D line D.15). It is the programmer's responsibility to insert the {SCALE,s,} couplet into the part program.

Examples: THREAD/LEAD,SCALE,1000000,.05,TURN

8.2 VARIABLES AND REGISTERS

A register is identified by a couplet containing the variable name and an index.

There shall be 99 general purpose registers which are referenced by VARBL,n. The values of registers 1 through 16 shall be saved by the implementor each time a subroutine is called, and shall be restored upon termination of each called subroutine. This allows each subroutine access to a temporary set of working registers.

There shall be a minimum of 99 sets of tool data registers. Each set shall contain 25 registers. These register sets are referenced by @TOOLVAR@. Refer to the TOOLNO/ command for the variable names.

There shall be 16 general purpose registers which are utilized to store the results of logical operations. These registers are referenced as FLAG,n. The values of the FLAG,n registers are either true or false. The values of flags 1 through 8 shall be saved by the implementor each time a subroutine is called, and shall be restored upon termination of each called subroutine. This allows each subroutine access to a temporary set of working registers.

There shall be 64 origin table register sets which are utilized to store the basic offsets, and 64 origin adjust table register sets which are used to store the secondary offsets. These registers are referenced by @ORG@ and @ORGADJ@, respectively. Refer to the ORIGIN/ and ORIGIN/ADJUST commands.

Some optional function sets (see Section 10.0) have additional registers associated with them. If Section 10.23, EXTENDED MATHEMATICAL OPERATIONS FUNCTION SET, is specified, there shall be 12 transform matrix registers available. If Section 10.26, BASIC PROBE FUNCTION SET, is specified, there shall be 4 probe register sets available. If Section 10.27, EXTENDED PROBE FUNCTION SET, is specified, there shall be 100 probe register sets and 12 transform matrix registers available.

It is the programmer's responsibility to initialize all registers prior to use. The contents of the registers are changed only by the execution of command records or operator intervention. This permits the passing of data from one program to the next.

The following conventions are used to define the syntax of variables and registers:

1. General variables and registers

FLAG(BE9),n = logical state of flag 'n'

VARBL(C27),n = contents of general purpose register 'n'

@ORG@,n = contents of an origin register associated
with origin set 'n'

@ORGADJ@,n = contents of an origin adjust register
associated with origin adjust set 'n'

@PRBREG@,n = contents of a probe register
associated with probe set 'n'

@TOOLVAR@,n = contents of a tool data register
associated with logical tool number 'n'

PRBSET(C14),n = collection of probe registers specified
by 'n'

The @...@ syntax signifies that one of the registers of the named register collection is to be chosen.

2. Triple registers

ORG(BF3),n = XORG, YORG, and ZORG of ORIGIN table entry 'n'

ORGADJ(BD3),n = XORGA, YORGA, and ZORGA of ORIGIN ADJUST table entry 'n'

PHIT(C05),n = XHIT, YHIT, and ZHIT of PRBSET,n

PNOM(C10),n = XNOM, YNOM, and ZNOM of PRBSET,n

XFRM(BCB),n = XTM, YTM, and ZTM values of column 'n' of the transform matrix XFRM

Non-integer indexes shall be rounded to the nearest integer. Within the command records defined in Section 8.3, a numerical value may be replaced by a variable identifier (e.g. VARBL,n) that contains the desired value. The index of the replacement variable may be replaced by another variable, but its index shall not be replaced by another variable (e.g. VARBL,VARBL,1 but not VARBL,VARBL,VARBL,1).

Using optional registers may decrease portability.

Mathematical operations may be performed with variables or constants as operands, but results of mathematical operations can only be variables and can not be constants.

Refer to figures 8-1A and 8-1B.

8.2.1 Origin Table Registers

The origin table triple register ORG(BF3),n comprises the following registers:

[XORG(BF4),n], [YORG(BF5),n], [ZORG(BF6),n]

The origin adjust table triple register ORGADJ(BD3),n comprises the following registers:

[XORGA(BD4),n], [YORGA(BD7),n], [ZORGA(BD8),n]

8.2.2 Probe Registers

The probe set PRBSET,0 is unique and always contains the results of a probing operation. The definition for PRBSET,0 is different from PRBSET,n where 'n' is greater than 0. PRBSET,0 comprises the following registers:

[XNOM(C2F),0], [YNOM(C37),0], [ZNOM(C3C),0],
[XHIT(C2D),0], [YHIT(C35),0], [ZHIT(C3A),0],
[ANOM(C47),0], [BNOM(C48),0], [CNOM(C49),0],
[AHIT(BD5),0], [BHIT(BD9),0], [CHIT(BDC),0],
[TOTDEV(C24),0], [PBSTAT(C1C),0]

The triple register set PNOM(C10),n comprises the following registers:

[XNOM(C2F),n], [YNOM(C37),n], [ZNOM(C3C),n]

The triple register set PHIT(C05),n comprises the following registers:

[XHIT(C2D),n], [YHIT(C35),n], [ZHIT(C3A),n]

The values within the probe set can only be specified as probe registers. For example, to specify the nominal value for the probed point in the X axis, the couplet, [XNOM,0] must be programmed.

The registers of probe register set PRBSET,0 are defined as follows:

[XNOM,0] is the nominal X axis value programmed with a single point destination record.

[YNOM,0] is the nominal Y axis value programmed with a single point destination record.

[ZNOM,0] is the nominal Z axis value programmed with a single point destination record.

[XHIT,0] is the actual surface contact point BCL X axis value when the probe hit occurred.

[YHIT,0] is the actual surface contact point BCL Y axis value when the probe hit occurred.

[ZHIT,0] is the actual surface contact point BCL Z axis value when the probe hit occurred.

[ANOM,0] is the nominal A axis value programmed with a ROTATE/ command.

[BNOM,0] is the nominal B axis value programmed with a ROTATE/ command.

[CNOM,0] is the nominal C axis value programmed with a ROTATE/ command.

[AHIT,0] is the actual A axis value when the probe hit occurred.

[BHIT,0] is the actual B axis value when the probe hit occurred.

[CHIT,0] is the actual C axis value when the probe hit occurred.

$$[TOTDEV,0] = ((([XHIT,n]-[XNOM,n])^2 + ([YHIT,n]-[YNOM,n])^2 + ([ZHIT,n]-[ZNOM,n])^2)^{0.5}$$

[PBSTAT,0] contains status information from the probing operation. The values for the [PBSTAT,0] register are:

- 0 - condition met within range 'r'
- 1 - condition not met
- 2 - condition met outside of range 'r'
- 3 - condition already exists

The probe registers [ANOM,0], [BNOM,0], [CNOM,0], [AHIT,0], [BHIT,0], and [CHIT,0] are unique to PRBSET,0. Only the register corresponding to the rotary axis on the machine contains valid data.

For the Basic Probe Function Set, the contents of probe register [TOTDEV,n] shall always be zero.

The probe sets PRBSET,1 through PRBSET,99 are defined with the probe registers as shown below:

[XNOM(C2F),n], [YNOM(C37),n], [ZNOM(C3C),n],
[XHIT(C2D),n], [YHIT(C35),n], [ZHIT(C3A),n],
[TOTDEV(C24),n], [PBSTAT(C1C),n]

The definitions for the registers are the same as above for the corresponding registers for PRBSET,0.

8.2.3 Transform Matrix Registers

The transform matrix triple register sets XFRM,0 through XFRM,3 comprise the following registers:

[XTM(C39),n], [YTM(C3D),n], [ZTM(C4A),n]

8.2.4 Tool Data Table Registers

Each tool data register set comprises the logical tool number and the 25 registers represented by @TOOLVAR@. Refer to the TOOLNO/TOOL command for the variable names and definitions.

The tool data table registers are accessed by the logical tool number.

For example:

XWEAR,6 = XWEAR value of logical tool number 6

GENERAL PURPOSE REGISTERS

VARBL,1	VARBL,2	VARBL,3	VARBL,4	VARBL,5		VARBL,97	VARBL,98	VARBL,99
---------	---------	---------	---------	---------	--	----------	----------	----------

FLAG REGISTERS

FLAG,1	FLAG,2	FLAG,3	FLAG,4	FLAG,5		FLAG,14	FLAG,15	FLAG,16
--------	--------	--------	--------	--------	--	---------	---------	---------

ORIGIN REGISTERS

ORIGIN TABLE

n_1	XORG, n_1	YORG, n_1	ZORG, n_1
n_2	XORG, n_2	YORG, n_2	ZORG, n_2
n_3	XORG, n_3	YORG, n_3	ZORG, n_3
n_{63}	XORG, n_{63}	YORG, n_{63}	ZORG, n_{63}
n_{64}	XORG, n_{64}	YORG, n_{64}	ZORG, n_{64}

ORG, n_i

ORIGIN ADJUST TABLE

n_1	XORGA, n_1	YORGA, n_1	ZORGA, n_1
n_2	XORGA, n_2	YORGA, n_2	ZORGA, n_2
n_3	XORGA, n_3	YORGA, n_3	ZORGA, n_3
n_{63}	XORGA, n_{63}	YORGA, n_{63}	ZORGA, n_{63}
n_{64}	XORGA, n_{64}	YORGA, n_{64}	ZORGA, n_{64}

ORGADJ, n_i

TRANSFORM REGISTERS

0:	XTM,0	YTM,0	ZTM,0
1:	XTM,1	YTM,1	ZTM,1
2:	XTM,2	YTM,2	ZTM,2
3:	XTM,3	YTM,3	ZTM,3

XFRM, i

FIGURE 8-1 A
BCL STORAGE REGISTERS, PAGE 1

PROBE REGISTERS

PRBSET, 0:	XNOM, 0	YNOM, 0	ZNOM, 0	XHIT, 0	YHIT, 0	ZHIT, 0	TOTDEV, 0	PBSTAT, 0	AND, 0	BNOM, 0	CNOM, 0	AHIT, 0	BHIT, 0	CHIT, 0
PRBSET, 1:	XNOM, 1	YNOM, 1	ZNOM, 1	XHIT, 1	YHIT, 1	ZHIT, 1	TOTDEV, 1	PBSTAT, 1						
PRBSET, 2:	XNOM, 2	YNOM, 2	ZNOM, 2	XHIT, 2	YHIT, 2	ZHIT, 2	TOTDEV, 2	PBSTAT, 2						
PRBSET, 3:	XNOM, 3	YNOM, 3	ZNOM, 3	XHIT, 3	YHIT, 3	ZHIT, 3	TOTDEV, 3	PBSTAT, 3						
PRBSET, 98:	XNOM, 98	YNOM, 98	ZNOM, 98	XHIT, 98	YHIT, 98	ZHIT, 98	TOTDEV, 98	PBSTAT, 98						
PRBSET, 99:	XNOM, 99	YNOM, 99	ZNOM, 99	XHIT, 99	YHIT, 99	ZHIT, 99	TOTDEV, 99	PBSTAT, 99						
	PNOM, i			PHIT, i										

TOOL DATA REGISTERS

TLINDX, 1:	n ₁	HOLDER	DELTA X	DELTA Z	FPN	LEAD	ORIENT	PILOT L	RADIUS	RPM	SIZE	TEETH	TIP LOC	TL ANGL	TL DEAD	TL LIFE	TL TIME	TL USER	TYPE	XL ENTH	XWEAR	XUR MAX	YL ENTH	ZL ENTH	ZWEAR	ZUR MAX
TLINDX, 2:	n ₂	HOLDER	DELTA X	DELTA Z	FPN	LEAD	ORIENT	PILOT L	RADIUS	RPM	SIZE	TEETH	TIP LOC	TL ANGL	TL DEAD	TL LIFE	TL TIME	TL USER	TYPE	XL ENTH	XWEAR	XUR MAX	YL ENTH	ZL ENTH	ZWEAR	ZUR MAX
TLINDX, 99:	n ₉₉	HOLDER	DELTA X	DELTA Z	FPN	LEAD	ORIENT	PILOT L	RADIUS	RPM	SIZE	TEETH	TIP LOC	TL ANGL	TL DEAD	TL LIFE	TL TIME	TL USER	TYPE	XL ENTH	XWEAR	XUR MAX	YL ENTH	ZL ENTH	ZWEAR	ZUR MAX
	n _i	@TOOLVAR@, n _i																								

FIGURE 8-1 B
BCL STORAGE REGISTERS, PAGE 2

8.3 RECORD DEFINITIONS

This section defines both Primary and Auxiliary command records. Primary commands are machine-independent commands (except where noted). Auxiliary commands are machine-dependent commands.

Primary command records may appear in BCL part programs, embedded subroutines, and external subroutines. Primary commands, except where noted, should not adversely affect portability.

Auxiliary command records shall not appear in BCL part programs or embedded subroutines but may appear in external subroutines. Auxiliary command records shall be accessed only through the PARTOP/ command or an appropriate error action shall occur. Auxiliary commands adversely affect portability if contained within BCL part programs, but should not when contained within external subroutines.

AIR(3F3)/[ON(47)]
[OFF(48)]

Primary command. Function Set 10.1.

[ON] turns on an air blast used for chip clearance.

[OFF] turns off an air blast used for chip clearance.

ARC/x1,y1,z1,i,j,k,x2,y2,z2

Primary command. Function Sets 10.1 and 10.12.

This command has a special binary representation. Refer to Section 6.2.1.2 for definition.

BARFED(BB9)/z1,z2

Auxiliary command. Function Set 10.11.

For turning machines.

BARFED/ is used to control optional mechanisms for advancing bar stock.

Bar stock is advanced to the point specified by 'z2', relative to the part program zero.

'z1' is the position of the cutoff face, in the part coordinate system, prior to execution of the BARFED/ command.

'z2' is the position of the cutoff face, in the part coordinate system, after execution of the BARFED/ command.

BRANCH(BBA)/{ FLAG(BE9),f,} label
{PNOT(C11),FLAG(BE9),f,}

Primary command. Function Set 10.24.

A BRANCH record transfers execution of the BCL program to the LABEL specified by the 'label' triplet. Each BRANCH record in a BCL program must have a companion LABEL record. If the specified LABEL is not contained within the program, then the numerical control unit system shall cause an appropriate error action to occur.

{FLAG,f} causes the branch to be taken if the flag is true. {PNOT,FLAG,f} causes the branch to be taken if the flag is false. The BCL command record following the BRANCH is executed if the logical condition is not satisfied.

The BRANCH causes a transfer only within the subroutine or mainline program in which it is encountered.

All BCL modal data remains the same as before the BRANCH.

BRANCH(BBA)/CGOTO(C41),i,label ...

Primary command. Function Set 10.24.

The index 'i' causes a branch to one of the LABEL records enumerated by the 'label' triplets which follow it. Each BRANCH record in a BCL program must have a companion LABEL record. If the specified LABEL is not contained within the program, then the numerical control unit system shall cause an appropriate error action to occur.

'i' is rounded to the nearest integer to obtain the index. If 'i' is zero, negative, or if there are insufficient parameters, this record shall be ignored.

The BRANCH causes a transfer only within the subroutine or mainline program in which it is encountered.

All BCL modal data remains the same as before the BRANCH.

Order: The parameters shall appear in the order shown.

CALC(BBB)/

Primary commands.

The description in this section applies to the CALC/ commands which follow.

CALC/ commands perform mathematical operations on one or more operands obtained from a source variable, register or constant and store the result in a destination variable or register. Sources shown as lower case letters represent variables, registers, or constants. Sources shown as upper case letters represent triple registers (refer to Section 8.2). Destinations shown as lower case letters represent variables or registers, but not constants. Destinations shown as upper case letters represent triple registers. Sources and destinations must be of compatible length, except that if the source(s) is a single value, variable, or register and the destination is of multiple length, the result shall fill every element of the destination.

Calculation errors shall cause an appropriate error action to occur. It is the programmer's responsibility to handle any possible errors.

The control builder is responsible for maintaining sufficient precision. Calculation error shall be less than the least significant bit for the scale factor specified by the RESOLN/n command.

Refer to figure 8-2.

```

CALC(BBB)/d,[PABS(BF7) ],a
[PACOS(BFA) ]
[PASIN(BFB) ]
[PATAN(BFC) ]
[PCOS(BFE) ]
[PNEG(COF) ]
[PSIN(C1A) ]
[PSQRT(C1B) ]
[PTAN(C1E) ]

```

Primary command. Function Set 10.22.

This command performs a mathematical operation on 'a' and stores the result in the variable or register specified by 'd'.

d = OPERATION (a) as follows:

OPERATION	PERFORMS	ERROR CONDITIONS
PABS	d = Absolute Value(a)	
PACOS	d = ARCCOS(a)	ABS(a) > 1
PASIN	d = ARCSIN(a)	ABS(a) > 1
PATAN	d = ARCTAN(a)	
	$-90^{\circ} < d < 90^{\circ}$	
PCOS	d = COS(a)	
PNEG	d = -a	
PSIN	d = SIN(a)	
PSQRT	d = Square Root(a)	a < 0
PTAN	d = TAN(a)	

Angles are in degrees.

Order: The parameters shall appear in the order shown.

```

CALC(BBB)/d,[PEXP(C01) ],a
[PEXPE(C02) ]
[PINTGR(C07) ]
[PLOG(C09) ]
[PLOGE(C0A) ]
[PROUND(C19) ]

```

Primary command. Function Set 10.22.

This command performs a mathematical operation on 'a' and stores the result in the variable or register specified by 'd'.

d = OPERATION (a) as follows:

OPERATION	PERFORMS	RESTRICTIONS
PEXP	d = 10^a	
PEXPE	d = e^a	
PINTGR	d = most positive integer $\leq a$	a $\geq$ 0
	d = most positive integer $\geq a$	a < 0
PLOG	d = $\text{LOG}_{10}(a)$	a > 0
PLOGE	d = $\text{LOG}_e(a)$	a > 0
PROUND	d = PINTGR(a + .5)	a $\geq$ 0
	d = PINTGR(a - .5)	a < 0

Order: The parameters shall appear in the order shown.

```

CALC(BBB)/d,[PADD(BF8)  ],a,b
              [PATAN2(BFD)]
              [PDIV(BFF) ]
              [PMOD(COC) ]
              [PMULT(COD) ]
              [PPower(C13)]
              [PSUB(C1D) ]

```

Primary command. Function Set 10.22.

This command performs a mathematical operation on 'a' and 'b' and stores the result in the variable or register specified by 'd'.

d = a OPERATION b as follows:

OPERATION	PERFORMS	FORMULA	ERROR CONDITIONS
PADD	Addition	$d = a + b$	
PATAN2	Inverse tangent $-180^\circ < d \leq 180^\circ$	$d = \text{ATAN2}(a,b)$	$a = b = 0$
PDIV	Division	$d = a / b$	$b = 0$
PMOD	Congruence	$d = a \text{ MOD } b$	a or b not integer
PMULT	Multiplication	$d = a * b$	
PPower	Exponentiation	$d = a^b$	
PSUB	Subtraction	$d = a - b$	

Order: The parameters shall appear in the order shown.

```
CALC(BBB)/D,CROSS(3),A,B
```

Primary command. Function Set 10.23.

The cross product of the vectors defined by 'A' crossed into 'B' is stored in the triple register specified by 'D'.

The values stored in 'A' and 'B' are treated as the components of two vectors and must be set up prior to this calculation by the part programmer.

Order: The parameters shall appear in the order shown.

```
CALC(BBB)/d,DOT(BE4),A,B
```

Primary command. Function Set 10.23.

The dot product of the vectors defined by 'A' and 'B' is stored in the variable or register specified by 'd'.

The values stored in 'A' and 'B' are treated as the components of two vectors and must be set up prior to this calculation by the part programmer.

Order: The parameters shall appear in the order shown.

```
CALC(BBB)/d,LENGTH(9),A
```

Primary command. Function Set 10.23.

The length of a vector defined by 'A' is stored in the variable or register specified by 'd'.

The three values stored in 'A' are treated as the components of a vector and must be set up prior to this calculation by the part programmer.

Order: The parameters shall appear in the order shown.

CALC(BBB)/D,UNIT(1E),A

Primary command. Function Set 10.23.

The components of a unit vector are stored in the triple register specified by 'D'. The unit vector is calculated by normalizing the components of the vector specified by 'A'.

The three values stored in 'A' are treated as the components of a vector and must be set up prior to this calculation by the part programmer.

Order: The parameters shall appear in the order shown.

CALC(BBB)/XFORM(C2C)

Primary commands.

The description in this section applies to the CALC/XFORM commands which follow.

CALC/XFORM commands calculate a rotation transformation matrix and store the result as the transform matrix. The transform matrix shall be applied to all destination points and tool axis components whenever SET/XFORM,ON is in effect. IF ORIGIN/ROTREF,ON is in effect, the transform matrix shall also be applied to the ORIGIN components.

At the start of the program, the transform matrix shall be initialized to the Identity matrix.

The part coordinate system in some situations may be rotated as well as translated in space. That is, the part to be machined may be skewed on the machine. The transform matrix, XFRM, allows the machining of a part that is skewed or not aligned with the machine coordinate system. The XFRM matrix is defined below:

$$\begin{array}{c|c|c|c|c|c|c|c|c}
 \begin{array}{c} X_r \\ Y_r \\ Z_r \end{array} & = & \begin{array}{c} X_{TM,0} \\ Y_{TM,0} \\ Z_{TM,0} \end{array} & + & \begin{array}{ccc} X_{TM,1} & X_{TM,2} & X_{TM,3} \\ Y_{TM,1} & Y_{TM,2} & Y_{TM,3} \\ Z_{TM,1} & Z_{TM,2} & Z_{TM,3} \end{array} & * & \begin{array}{c} X_p \\ Y_p \\ Z_p \end{array}
 \end{array}$$

Coordinates of Resultant BCL point Translation Vector Rotation Matrix Coordinates of BCL Part program point

Xp, Yp, and Zp are the coordinates of a point represented in the part coordinate system.

Xr, Yr, and Zr are the coordinates of a point in the part program represented in the resultant BCL coordinate system. This resultant BCL coordinate system is the part coordinate system aligned with the machine axes when all rotary axes are at zero.

The origin of the resultant coordinate system is translated from the machine coordinate origin by the ORIGIN/ and ORIGIN/ADJUST vectors. Thus, the BCL part coordinate data is mathematically manipulated in the order shown below:

Obtain the resultant BCL coordinate by applying the contents of the XFRM matrix to the programmed part coordinates.

Obtain the machine coordinates by transforming the resultant BCL coordinates to the machine coordinates.

CALC(BBB)/XFORM(C2C),A,B

Primary command. Function Set 10.23.

The rotation matrix is calculated from a pair of vectors. The resultant matrix is one which causes the vector specified by triple register 'A' to be rotated about the part origin into the vector specified by triple register 'B'.

The elements of the translation vector are forced to zero.

Order: The parameters shall appear in the order shown.

CALC(BBB)/XFORM(C2C),PRBSET(C14),a,PRBSET(C14),b,PRBSET(C14),c

Primary command. Function Set 10.23.

The translation vector and the rotation matrix are calculated based upon the nominal and actual hit coordinates for the three points designated by 'a', 'b', and 'c'.

The translation vector is obtained by subtracting the elements of 'PNOM,a' from the elements of 'PHIT,a'.

The PNOM array is then translated to superimpose 'PNOM,a' on 'PHIT,a' by adding the elements of the translation vector to the corresponding elements of the three PNOM points.

The rotation matrix is then obtained in two steps. First, the vector from 'PNOM,a' to 'PNOM,b' is rotated parallel to the vector from 'PHIT,a' to 'PHIT,b'. Second, the plane defined by the PHIT points is rotated about the line from 'PHIT,a' to 'PHIT,b' to bring the point 'PNOM,c' into the plane defined by the PHIT points.

Order: The parameters shall appear in the order shown.

**CALC(BBB)/XFORM(C2C),[XAXIS(54)],a
[YAXIS(55)]
[ZAXIS(56)]**

Primary command. Function Set 10.23.

The transform matrix is calculated based upon the specified axis and 'a' degrees. The resultant matrix is one that causes rotation about the specified part axis by 'a' degrees.

Order: The parameters shall appear in the order shown.

CASE	EXAMPLE BCL COMMAND	OPERATIONS DONE
1.	SCALAR = SCALAR + SCALAR CALC / XNDM,1, PADD, VARBL,1, VARBL,2	XNDM,1 = VARBL,1 + VARBL,2
2.	VECTOR = VECTOR + VECTOR CALC / PNDM,1, PADD, PNDM,2, PHIT,2	XNDM,1 = XNDM,2 + XHIT,2 YNDM,1 = YNDM,2 + YHIT,2 ZNDM,1 = ZNDM,2 + ZHIT,2
3.	PROBE SET = PROBE SET + PROBE SET CALC / PRBSET,1, PADD, PRBSET,2, PRBSET,3	XNDM,1 = XNDM,2 + XNDM,3 YNDM,1 = YNDM,2 + YNDM,3 ZNDM,1 = ZNDM,2 + ZNDM,3 XHIT,1 = XHIT,2 + XHIT,3 YHIT,1 = YHIT,2 + YHIT,3 ZHIT,1 = ZHIT,2 + ZHIT,3 TOTDEV,1 = TOTDEV,2 + TOTDEV,3
4.	VECTOR = SCALAR + VECTOR VECTOR = VECTOR + SCALAR CALC / PNDM,1, PADD, VARBL,1 PHIT,2	XNDM,1 = VARBL,1 + XHIT,2 YNDM,1 = VARBL,1 + YHIT,2 ZNDM,1 = VARBL,1 + ZHIT,2
5.	PROBE SET = PROBE SET + SCALAR PROBE SET = SCALAR + PROBE SET CALC / PRBSET,1, PADD, PRBSET,2, VARBL,3	XNDM,1 = XNDM,2 + VARBL,3 YNDM,1 = YNDM,2 + VARBL,3 ZNDM,1 = ZNDM,2 + VARBL,3 XACT,1 = XHIT,2 + VARBL,3 YACT,1 = YHIT,2 + VARBL,3 ZACT,1 = ZHIT,2 + VARBL,3 TOTDEV,1 = TOTDEV,2 + VARBL,3

FIGURE 8-2

CALCULATION EXPANSION

The arguments supplied for mathematical operations must be of compatible type and length. For the 5 cases shown, any scalar operation supplied with non-scalar arguments will be repeated for each component in the argument. Note that for the last 2 cases, the scalar may appear in either position.

CALSUB(476)/label

Primary command. Function Set 10.25.

Calls the BCL subroutine specified by the 'label' triplet. Subroutines called by the CALSUB/ command shall not contain Auxiliary commands (see PARTOP/).

The current program location of this CALSUB/ command is saved and execution is transferred to the start of the subroutine. Modal values in effect at the time of the CALSUB/ command shall continue into the subroutine. Modal values changed within a subroutine shall be in effect after returning to the calling program.

Subroutines may be nested eight deep, including the first subroutine.

Subroutines may be embedded within the part program or be separate part program files.

Subroutines embedded in the BCL part program file shall have precedence over separate part program file subroutines.

Embedded subroutines are defined by a DEFSUB/ - ENDSUB pair of command records. The subroutines shall be located immediately after the RESOLN/n command record and immediately before the part program data in the BCL file.

A part program file may be called as a subroutine. The subroutine name is the part program identifier as defined in the PARTNO record. The END or FINI command record shall cause execution to resume at the command record after the CALSUB/ command in the calling program.

A CALSUB/ to an unidentified subroutine shall cause an appropriate error action to occur.

**CATCHR(BBC)/[IN(30)]
[OUT(31)]**

Auxiliary command. Function Set 10.11.

For turning machines.

CATCHR/ is used to control optional mechanisms for a parts catcher.

[IN] positions the catching bin into position for catching parts.

[OUT] retracts the catching bin.

CHECK(3FF)/GAGE(144)

Auxiliary command. Function Set 10.26.

Calibrates a tool gaging device. The actual action performed is implementor dependent.

CHECK(3FF)/PART(A6)

Primary command. Function Sets 10.26 and 10.27.

Function Set 10.26, BASIC PROBE FUNCTION SET, provides a basic set of probe functions for machines with limited probing requirements.