

Binary Numbers and the Binary Coded Decimal System. In the binary number system, each digit can take on either of two values, 0 or 1. The meaning of consecutive digits in the binary system is based on the number 2 raised to successive powers. Starting from the right, the first digit is 2^0 (which equals 1), the second digit is 2^1 (which equals 2), the third is 2^2 (which equals 4), the fourth is 2^3 (which equals 8), and so forth. The two numbers, 0 or 1, in successive digit positions, indicate the presence or absence of the value. For example, the binary number 0101 is equal to the decimal number 5. The conversion from binary to decimal operates as follows:

$$(0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (0 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) \\ = 4 + 1 = 5$$

Conversion of the 10 digits in the decimal number system into binary numbers is shown in Table 6.6. Four binary digits are required to represent the ten single-digit numbers in decimal. Of course, the numerical data required in NC includes large decimal values; for example, the coordinate position $x = 1250$ mm. To encode the decimal value 1250 in the binary number system requires a total of 11 digits: 10011100010. Another problem with the binary number system is the coding of decimal fractions, for example, feed = 0.085 mm/rev.

To deal with these problems in NC, a combination of the binary and decimal number systems has been adopted, called the *binary-coded decimal* (BCD) system. In this coding scheme, each of the ten digits (0-9) in the decimal system is coded as a four-digit binary number, and these binary numbers are added in sequence as in the decimal number system. For example, the decimal value 1250 would be coded in BCD as follows:

Number sequence	Binary number	Decimal value
First	0001	1000
Second	0010	200
Third	0101	50
Fourth	0000	0
Sum		1250

EIA and ISO Coding Standards. In addition to numerical values, the NC coding system must also provide for alphabetical characters and other symbols. Eight binary digits are used to represent all of the characters required for NC part programming. There are two standard coding systems currently used in NC: (1) the Electronics Industry Association (EIA) and (2) the International Standards Organization (ISO). The Electronics Industry Association system is known as EIA RS-244-B. The ISO code was originally developed as the American Standard Code for Information Interchange (ASCII) and has been adopted by ISO as its NC standard. The complete listings of EIA and ISO (ASCII) codes for NC are shown in Table 6.7. Many NC controllers are capable of reading either code.

TABLE 6.6 Comparison of Binary and Decimal Numbers

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

Both EIA and ISO coding schemes were developed when punched tape was the pre-dominant medium for storing NC part programs. Although punched tape has been largely superseded by more modern media, it is still widely used in industry, if only for backup storage. To ensure the correctness of the punched tape, the eight binary digits in the EIA and ISO codes include a *parity check*. Here's how the parity check works, explained here for the EIA code. In the EIA system, the tape reader is instructed to count an odd number of holes across the width of the tape. Whenever the particular number or symbol being punched requires an even number of holes, an extra hole is punched in column 5, hence making the total an odd number. For example, the decimal number 5 is coded by means of holes in columns 1 and 3. Since this is an even number of holes, a parity hole would be added. The decimal 7 requires an odd number of holes (in columns 1, 2, and 3), so no parity hole is needed. The parity check helps to ensure that the tape punch mechanism has perforated a complete hole in all required positions. If the tape reader counts an even number of holes, then a signal is issued that a parity error has occurred.

The difference between the EIA and ISO systems is that the parity check in the ISO code is an even number of holes, called an *even parity*. The EIA system uses an *odd parity*. Also, whereas the parity hole is in the fifth-digit position in the EIA coding system, it is in the eighth position in the ISO system. These differences can be seen in Table 6.7.

How Instructions Are Formed. A binary digit is called a *bit*. In punched tape, the values 0 or 1 are represented by the absence or presence of a hole in a certain row and column position (rows run across the tape; columns run lengthwise along the tape). Out of one row of bits a character is formed. A *character* is a combination of bits representing a numerical digit (0-9), an alphabetical letter (A-Z), or a symbol (Table 6.7). Out of a sequence of characters, a word is formed. A *word* specifies a detail about the operation, such as *x*-position, *y*-position, feed rate, or spindle speed. Out of a collection of words, a block is formed. A *block* is one complete NC instruction. It specifies the destination for the move, the speed and feed of the cutting operation, and other commands that determine explicitly what the machine tool will do. For example, an instruction block for a two-axis NC milling machine would likely include the *x*- and *y*-coordinates to which the machine table should be moved, the type of motion to be performed (linear or circular interpolation), the rotational speed of the milling cutter, and the feed rate at which the milling operation should be performed. Instruction blocks are separated by an end-of-block (EOB) symbol (a hole in column 8 in the EIA standard or holes in columns 2 and 4 in the ISO standard, as in Table 6.7).

The essential information in a part program is conveyed to the MCU by means of words that specify coordinates, feeds and speeds, tooling, and other commands necessary to operate the machine tool. Given the variety of machine tool types and the many different companies that build NC machine tools and MCUs, it is no surprise that several different formats have been developed over the years to specify words within an instruction block. These are often referred to as *tape formats*, because they were developed for punched tapes. More generally, they are known as *block formats*. At least five block formats have been developed [8]; these are briefly described in Table 6.8, with two lines of code for the drilling sequence shown in Figure 6.12.

The word address format with TAB separation and variable word order has been standardized by EIA as RS-274. It is the block format used on all modern controllers and is the format we will discuss here. It is usually referred to simply as the word address format even though it has been enhanced by tab separation and variable word order. Common letter prefixes used in the word address format are defined in Table 6.9.

TABLE 6.8 Five Block Formats Used in NC Programming

Block Format (Tape Format)	Example for Figure 6.12
<i>Fixed sequential format.</i> This format was used on many of the first commercially available NC machines. Each instruction block contains five words specified in only numerical data and in a very fixed order.	00100070000300003 00200070000600003
<i>Fixed sequential format with TAB ignored.</i> This is the same as the fixed sequential format except that TAB codes are used to separate the words for easier reading by humans.	001 00 07000 03000 03 002 00 07000 06000 03
<i>Tab sequential format.</i> This is the same as the preceding format except that words with the same value as in the preceding block can be omitted in the sequence.	001 00 07000 03000 03 002 00 06000
<i>Word address format.</i> This format uses a letter prefix to identify the type of word. See Table 6.9 for definition of prefixes. Repeated words can be omitted. The words run together, which makes the code difficult to read (for humans).	N001G00X07000Y03000M03 N002Y06000
<i>Word address format with TAB separation and variable word order.</i> This is the same format as the previous, except that words are separated by TABs, and the words in the block can be listed in any order. See Table 6.9 for definition of letter prefixes.	N001 G00 X07000 Y03000 M03 N002 Y06000

Note: Examples indicate point-to-point moves to two hole locations in Figure 6.12.

Words in an instruction block are intended to convey all of the commands and data needed for the machine tool to execute the move defined in the block. The words required for one machine tool type may differ from those required for a different type; for example, turning requires a different set of commands than milling. The words in a block are usually given in the following order (although the word address format allows variations in the order):

- sequence number (N-word)
- preparatory word (G-word); see Table 6.10 for definition of G-words
- coordinates (X-, Y-, Z-words for linear axes, A-, B-, C-words for rotational axes)
- feed rate (F-word)

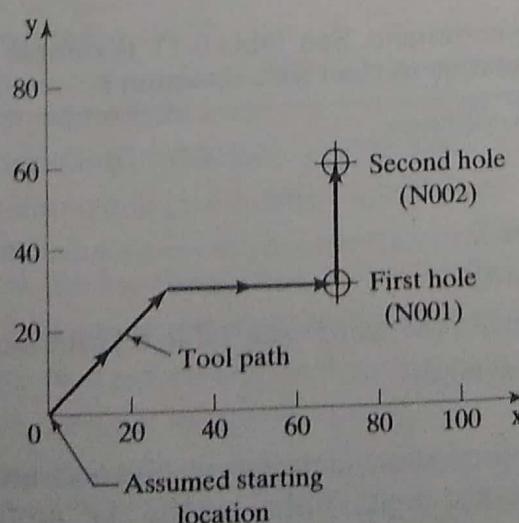


Figure 6.12 Example drilling sequence for block formats described in Table 6.8. Dimensions are in millimeters.

TABLE 6.9 Common Word Prefixes Used in Word Address Format

Word Prefix	Example	Function
N	N01	Sequence number; identifies block of instruction. From one to four digits can be used.
G	G21	Preparatory word; prepares controller for instructions given in the block. See Table 6.10. There may be more than one G-word in a block. (Example specifies that numerical values are in millimeters.)
X, Y, Z	X75.0	Coordinate data for three linear axes. Can be specified in either inches or millimeters. (Example defines x-axis value as 75 mm.)
U, W	U25.0	Coordinate data for incremental moves in turning in the x- and z-directions, respectively. (Example specifies an incremental move of 25 mm in the x-direction.)
A, B, C	A90.0	Coordinate data for three rotational axes. A is the rotational axis about x-axis; B rotates about y-axis; and C rotates about z-axis. Specified in degrees of rotation. (Example defines 90° of rotation about x-axis.)
R	R100.0	Radius of arc; used in circular interpolation. (Example defines radius = 100 mm for circular interpolation.) The R-code can also be used to enter cutter radius data for defining the tool path offset distance from the part edge.
I, J, K	I32 J67	Coordinate values of arc center, corresponding to x-, y-, and z-axes, respectively; used in circular interpolation. (Example defines center of arc for circular interpolation to be at x = 32 mm and y = 67 mm.)
F	G94 F40	Feed rate per minute or per revolution in either inches or millimeters, as specified by G-words in Table 6.10. (Example specifies feed rate = 40 mm/min in milling or drilling operation.)
S	S0800	Spindle rotation speed in revolutions per minute, expressed in four digits. For some machines, spindle rotation speed is expressed as a percentage of maximum speed available on machine, expressed in two digits.
T	T14	Tool selection, used for machine tools with automatic tool changers or tool turrets. (Example specifies that the cutting tool to be used in the present instruction block is in position 14 in the tool drum.)
D	D05	Tool diameter word used in contouring moves for offsetting the tool from the workpart by a distance stored in the indicated register, usually the distance is the cutter radius. (Example indicates that the radius offset distance is stored in offset register number 05 in the controller.)
P	P05 R15.0	Used to store cutter radius data in offset register number 05. (Example indicates that a cutter radius value of 15.0 mm is to be stored in offset register 05.)
M	M03	Miscellaneous command. See Table 6.11. (Example commands the machine to start spindle rotation in clockwise direction.)

Note: Dimensional values in the examples are specified in millimeters.

- spindle speed (S-word)
- tool selection (T-word)
- miscellaneous command (M-word); see Table 6.11 for definition of M-words
- end-of-block (EOB symbol)

G-words and M-words require some elaboration. G-words are called preparatory words. They consist of two numerical digits (following the "G" prefix in the word address for

mat) that prepare the MCU for the instructions and data contained in the block. For example, G02 prepares the controller for clockwise circular interpolation, so that the subsequent data in the block can be properly interpreted for this type of move. In some cases, more than one G-word is needed to prepare the MCU for the move. Most of the common G-words are presented in Table 6.10. While G-words have been standardized in the machine tool industry, there are sometimes deviations for particular machines. For instance, there are several differences between milling and turning type machines; these are identified in Table 6.10.

TABLE 6.10 Common G-words (Preparatory Word)

G-word	Function
G00	Point-to-point movement (rapid traverse) between previous point and endpoint defined in current block. Block must include x-y-z coordinates of end position.
G01	Linear interpolation movement. Block must include x-y-z coordinates of end position. Feed rate must also be specified.
G02	Circular interpolation, clockwise. Block must include either arc radius or arc center; coordinates of end position must also be specified.
G03	Circular interpolation, counterclockwise. Block must include either arc radius or arc center; coordinates of end position must also be specified.
G04	Dwell for a specified time.
G10	Input of cutter offset data, followed by a P-code and an R-code.
G17	Selection of x-y plane in milling.
G18	Selection of x-z plane in milling.
G19	Selection of y-z plane in milling.
G20	Input values specified in inches.
G21	Input values specified in millimeters.
G28	Return to reference point.
G32	Thread cutting in turning.
G40	Cancel offset compensation for cutter radius (nose radius in turning).
G41	Cutter offset compensation, left of part surface. Cutter radius (nose radius in turning) must be specified in block.
G42	Cutter offset compensation, right of part surface. Cutter radius (nose radius in turning) must be specified in block.
G50	Specify location of coordinate axis system origin relative to starting location of cutting tool. Used in some lathes. Milling and drilling machines use G92.
G90	Programming in absolute coordinates.
G91	Programming in incremental coordinates.
G92	Specify location of coordinate axis system origin relative to starting location of cutting tool. Used in milling and drilling machines and some lathes. Other lathes use G50.
G94	Specify feed per minute in milling and drilling.
G95	Specify feed per revolution in milling and drilling.
G98	Specify feed per minute in turning.
G99	Specify feed per revolution in turning.

Note: Some G-words apply to milling and/or drilling only, whereas others apply to turning only.

168
TABLE 6.11 Common M-words Used in Word Address Format

M-word	Function
M00	Program stop; used in middle of program. Operator must restart machine.
M01	Optional program stop; active only when optional stop button on control panel has been depressed.
M02	End of program. Machine stop.
M03	Start spindle in clockwise direction for milling machine (forward for turning machine).
M04	Start spindle in counterclockwise direction for milling machine (reverse for turning machine).
M05	Spindle stop.
M06	Execute tool change, either manually or automatically. If manually, operator must restart machine. Does not include selection of tool, which is done by T-word if automatic, by operator if manual.
M07	Turn cutting fluid on flood.
M08	Turn cutting fluid on mist.
M09	Turn cutting fluid off.
M10	Automatic clamping of fixture, machine slides, etc.
M11	Automatic unclamping.
M13	Start spindle in clockwise direction for milling machine (forward for turning machine) and turn on cutting fluid.
M14	Start spindle in counterclockwise direction for milling machine (reverse for turning machine) and turn on cutting fluid.
M17	Spindle and cutting fluid off.
M19	Turn spindle off at oriented position.
M30	End of program. Machine stop. Rewind tape (on tape-controlled machines).

M-words are used to specify miscellaneous or auxiliary functions that are available on the machine tool. Examples include starting the spindle rotation, stopping the spindle for a tool change, and turning the cutting fluid on or off. Of course, the particular machine tool must possess the function that is being called. Many of the common M-words are explained in Table 6.11. Miscellaneous commands are normally placed at the end of the block.

6.5.2 Manual Part Programming

In manual part programming, the programmer prepares the NC code using the low-level machine language previously described. The program is either written by hand on a form from which a punched tape or other storage media is subsequently coded, or it is entered directly into a computer equipped with NC part programming software, which writes the program onto the storage media. In any case, the part program is a block-by-block listing of the machining instructions for the given job, formatted for the particular machine tool to be used.

Manual part programming can be used for both point-to-point and contouring jobs. It is most suited for point-to-point machining operations such as drilling. It can also be used for simple contouring jobs, such as milling and turning when only two axes are in-

volved. However, for complex three-dimensional machining operations, there is an advantage in using computer-assisted part programming.

Instructions in Word Address Format. Instructions in word address format consist of a series of words, each identified by a prefix label. In our coverage, statements are illustrated with dimensions given in millimeters. The values are expressed in four digits including one decimal place. For example, X020.0 means $x = 20.0$ mm. It should be noted that many CNC machines use formats that differ from ours, and so the instruction manual for each particular machine tool must be consulted to determine its own proper format. Our format is designed to convey principles and for easy reading.

In preparing the NC part program, the part programmer must initially define the origin of the coordinate axes and then reference the succeeding motion commands to this axis system. This is accomplished in the first statement of the part program. The directions of the x -, y -, and/or z -axes are predetermined by the machine tool configuration, but the origin of the coordinate system can be located at any desired position. The part programmer defines this position relative to some part feature that can be readily recognized by the machine operator. The operator is instructed to move the tool to this position at the beginning of the job. With the tool in position, the G92 code is used by the programmer to define the origin as follows:

G92 X0 Y-050.0 Z010.0

where the x , y , and z values specify the coordinates of the tool location in the coordinate system; in effect, this defines the location of the origin. In some CNC lathes and turning centers, the code G50 is used instead of G92. Our x , y , and z values are specified in millimeters, and this would have to be explicitly stated. Thus, a more-complete instruction block would be the following:

G21 G92 X0 Y-050.0 Z010.0

where the G21 code indicates that the subsequent coordinate values are in millimeters. Motions are programmed by the codes G00, G01, G02, and G03. G00 is used for a point-to-point rapid traverse movement of the tool to the coordinates specified in the command; for example,

G00 X050.0 Y086.5 Z100.0

specifies a rapid traverse motion from the current location to the location defined by the coordinates $x = 50.0$ mm, $y = 86.5$ mm, and $z = 100.0$ mm. This command would be appropriate for NC drilling machines in which a rapid move is desired to the next hole location, with no specification on the tool path. The velocity with which the move is achieved in rapid traverse mode is set by parameters in the MCU and is not specified numerically in the instruction block. The G00 code is not intended for contouring operations.

Linear interpolation is accomplished by the G01 code. This is used when it is desired for the tool to execute a contour cutting operation along a straight line path. For example, the command

G01 G94 X050.0 Y086.5 Z100.0 F40 S800

specifies that the tool is to move in a straight line from its current position to the location defined by $x = 50.0$ mm, $y = 86.5$ mm, and $z = 100.0$ mm, at a feed rate of 40 mm/min and spindle speed of 800 rev/min.

The G02 and G03 codes are used for circular interpolation, clockwise and counter-clockwise, respectively. As indicated in Table 6.1, circular interpolation on a milling machine is limited to one of three planes, x - y , x - z , or y - z . The distinction between clockwise and counterclockwise is established by viewing the plane from the front view. Selection of the desired plane is accomplished by entering one of the codes, G17, G18, or G19, respectively. Thus, the instruction

G02 G17 X088.0 Y040.0 R028.0 F30

moves the tool along a clockwise circular trajectory in the x - y plane to the final coordinates defined by $x = 88$ mm and $y = 40$ mm at a feed rate of 30 mm/min. The radius of the circular arc is 28 mm. The path taken by the cutter from an assumed starting point ($x = 40$, $y = 60$) is illustrated in Figure 6.13.

In a point-to-point motion statement (G00), it is usually desirable to position the tool so that its center is located at the specified coordinates. This is appropriate for operations such as drilling, in which a hole is to be positioned at the coordinates indicated in the statement. But in contouring motions, it is almost always desirable that the path followed by the center of the tool be separated from the actual surface of the part by a distance equal to the cutter radius. This is shown in Figure 6.14 for profile milling the outside edges of a rectangular part in two dimensions. For a three-dimensional surface, the shape of the end of the cutter would also have to be considered in the offset computation. This tool path compensation is called the *cutter offset*, and the calculation of the correct coordinates of the endpoints of each move can be time consuming and tedious for the part programmer. Modern CNC machine tool controllers perform these cutter offset calculations automatically when the programmer uses the G40, G41, and G42 codes. The G40 code is used to cancel the cutter offset compensation. The G41 and G42 codes invoke the cutter offset compensation of the tool path on the left- or right-hand side of the part, respectively. The left- and right-hand sides are defined according to the tool path direction. To illustrate, in the rectangular part

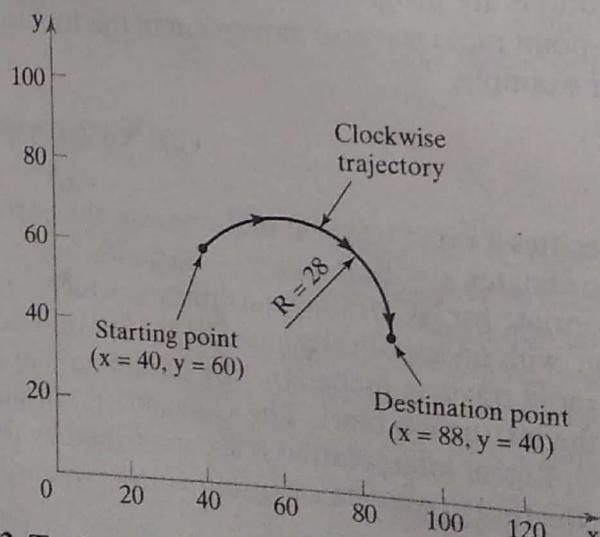


Figure 6.13 Tool path in circular interpolation for the statement:
G02 G17 X088.0 Y040.0 R028.0. Units are millimeters.

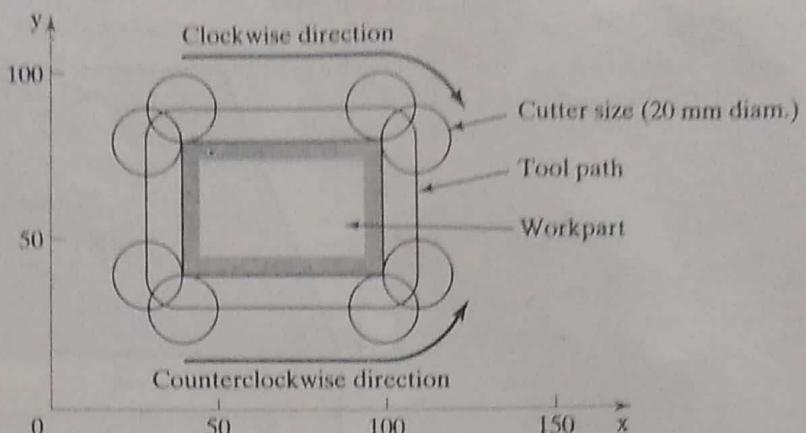


Figure 6.14 Cutter offset for a simple rectangular part. The tool path is separated from the part perimeter by a distance equal to the cutter radius. To invoke cutter offset compensation, the G41 code is used to follow the clockwise path, which keeps the tool on the left-hand side of the part. G42 is used to follow the counterclockwise path, which keeps the tool on the right-hand side of the part.

in Figure 6.14, a clockwise tool path around the part would always position the tool on the left-hand side of the edge being cut, so a G41 code would be used to compute the cutter offset compensation. By contrast, a counterclockwise tool path would keep the tool on the right-hand side of the part, so G42 would be used. Accordingly, the instruction for profile milling the bottom edge of the part, assuming that the cutter begins along the bottom left corner, would read:

G42 G01 X100.0 Y040.0 D05

where D05 refers to the cutter radius value stored in MCU memory. Certain registers are reserved in the control unit for these cutter offset values. The D-code references the value contained in the identified register. D05 indicates that the radius offset distance is stored in the number 5 offset register in the controller. This data can be entered into the controller in either of two ways: (1) as manual input or (2) as an instruction in the part program. Manual input is more flexible because the tooling used to machine the part may change from one setup to the next. At the time the job is run, the operator knows which tool will be used, and the data can be loaded into the proper register as one of the steps in the setup. When the offset data is entered as a part program instruction, the statement has the form:

G10 P05 R10.0

where G10 is a preparatory word indicating that cutter offset data will be entered; P05 indicates that the data will be entered into offset register number 05; and R10.0 is the radius value, here 10.0 mm.

Some Part Programming Examples. To demonstrate manual part programming, we present two examples using the sample part shown in Figure 6.15. The first example is a point-to-point program to drill the three holes in the part. The second example is a two-axis contouring program to accomplish profile milling around the periphery of the part.

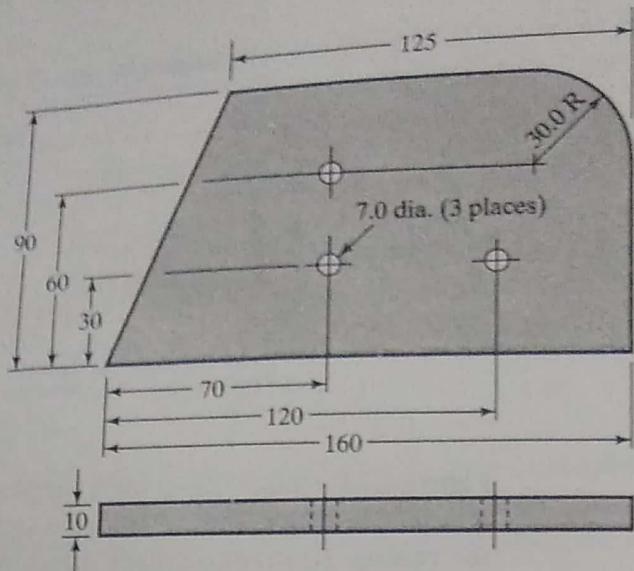


Figure 6.15 Sample part to illustrate NC part programming. Dimensions are in millimeters. General tolerance = ± 0.1 mm. Work material is a machinable grade of aluminum.

EXAMPLE 6.1 Point-to-Point Drilling

This example presents the NC part program in word address format for drilling the three holes in the sample part shown in Figure 6.15. We assume that the outside edges of the starting workpart have been rough cut (by jig sawing) and are slightly oversized for subsequent profile milling. The three holes to be drilled in this example will be used to locate and fixture the part for profile milling in the following example. For the present drilling sequence, the part is gripped in place so that its top surface is 40 mm above the surface of the machine tool table to provide ample clearance beneath the part for hole drilling. We will define the x -, y -, and z -axes as shown in Figure 6.16. A 7.0-mm diameter drill, corresponding to the specified hole size, has been chucked in the CNC drill press. The drill will be operated at a feed of 0.05 mm/rev and a spindle speed of 1000 rev/min (corresponding to a surface speed of about 0.37 m/sec, which is slow for the aluminum work material). At the beginning of the job, the drill point will be positioned at a target point located at $x = 0$, $y = -50$, and $z = +10$ (axis units are millimeters). The program begins with the tool positioned at this target point.

NC Part Program Code

```

N001 G21 G90 G92 X0 Y-050.0 Z010.0;
N002 G00 X070.0 Y030.0;
N003 G01 G95 Z-15.0 F0.05 S1000 M03;
N004 G01 Z010.0;
N005 G00 Y060.0;
N006 G01 G95 Z-15.0 F0.05;
N007 G01 Z010.0;

```

Comments

Define origin of axes.	
Rapid move to first hole location.	
Drill first hole.	
Retract drill from hole.	
Rapid move to second hole location.	
Drill second hole.	
Retract drill from hole.	

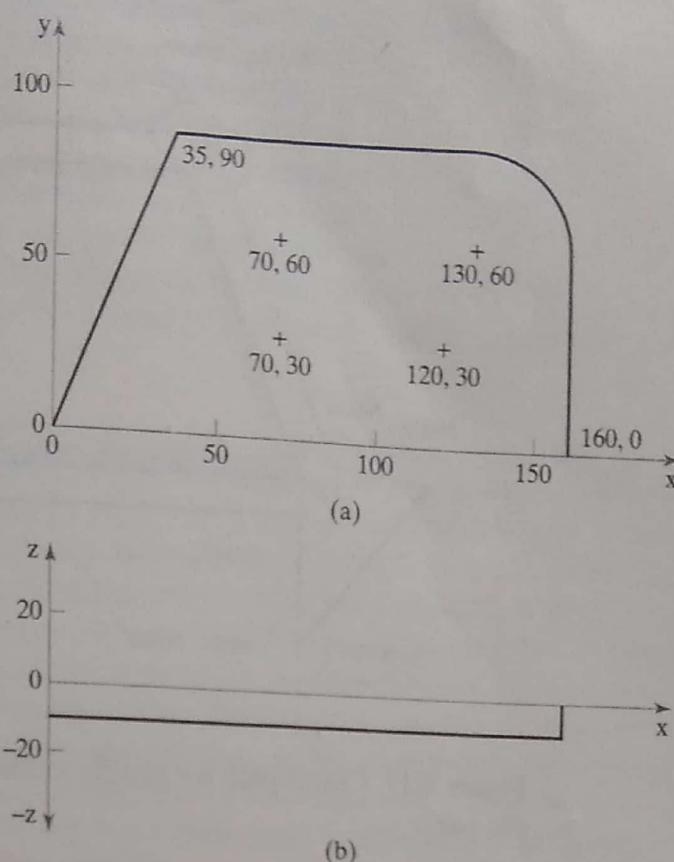


Figure 6.16 Sample part aligned relative to (a) x - and y -axes, and (b) z -axis. Coordinates are given for significant part features in (a).

N008 G00 X120.0 Y030.0;	Rapid move to third hole location.
N009 G01 G95 Z-15.0 F0.05;	Drill third hole.
N010 G01 Z010.0;	Retract drill from hole.
N011 G00 X0 Y-050.0 M05;	Rapid move to target point.
N012 M30;	End of program, stop machine.

EXAMPLE 6.2 Two-Axis Milling

The three holes drilled in the previous example can be used for locating and holding the workpart to completely mill the outside edges without re-fixture. The axis coordinates are shown in Figure 6.16 (same coordinates as in the previous drilling sequence). The part is fixture so that its top surface is 40 mm above the surface of the machine tool table. Thus, the origin of the axis system will be 40 mm above the table surface. A 20-mm diameter end mill with four teeth will be used. The cutter has a side tooth engagement length of 40 mm. Throughout the machining sequence, the bottom tip of the cutter will be positioned 25 mm below the part top surface, which corresponds to $z = -25$ mm. Since the part is 10 mm thick, this z -position will allow the side cutting edges of the milling cutter to cut the full thickness of the part during profile milling. The cutter will be operated at a spindle speed = 1000 rev/min (which corresponds to a surface speed of about 1.0 m/sec) and a feed rate = 50 mm/min (which corresponds to 0.20 mm/tooth). The tool path to be followed by the cutter is shown

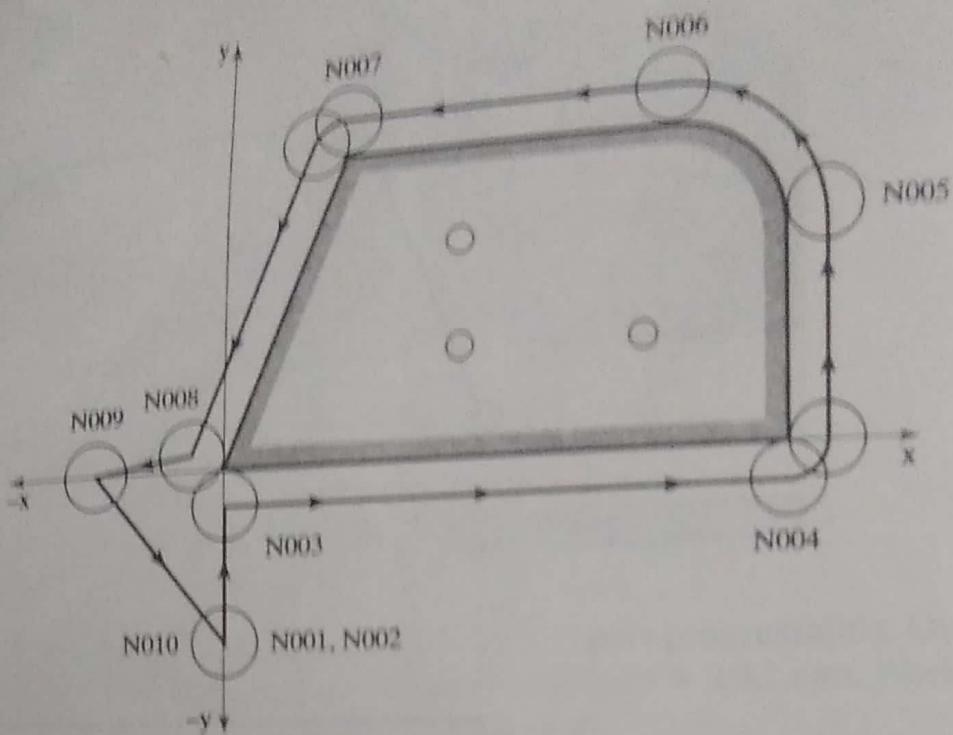


Figure 6.17 Cutter path for profile milling outside perimeter of sample part.

in Figure 6.17, with numbering that corresponds to the sequence number in the program. Cutter diameter data has been manually entered into offset register 05. At the beginning of the job, the cutter will be positioned so that its center tip is at a target point located at $x = 0$, $y = -50$, and $z = +10$. The program begins with the tool positioned at this location.

NC Part Program Code

```

N001 G21 G90 G92 X0 Y-050.0 Z010.0;
N002 G00 Z-025.0 S1000 M03;
N003 G01 G94 G42 Y0 D05 F40;
N004 G01 X160.0;
N005 G01 Y060.0;
N006 G17 G03 X130.0 Y090.0 R030.0;
N007 G01 X035.0;
N008 G01 X0 Y0;
N009 G40 G00 X-040.0 M05;
N010 G00 X0 Y-050.0;
N011 M30;
```

Comments

Define origin of axes.	N001
Rapid to cutter depth, turn spindle on.	N002
Engage part, start cutter offset.	N003
Mill lower part edge.	N004
Mill right straight edge.	N005
Circular interpolation around arc.	N006
Mill upper part edge.	N007
Mill left part edge.	N008
Rapid exit from part, cancel offset.	N009
Rapid move to target point.	N010
End of program, stop machine.	N011

6.5.3 Computer-Assisted Manufacturing

CNC Programming.

I ① Milling & Drilling Programming

① N5 G92 X-1.000 Y1.000 Z1.000

G92 Programmed offset of reference point (tool change position)

X-1.000 tool set at 1.000 to the left of part

Y1.000 tool set at 1.000 above the top edge of part

Z1.000 the end of the cutter is 1.000 above the top surface of the part

② N10 G20 G90

G20 inch data input

G90 absolute programming mode

③ N15 M06 T01

M06 tool change command

PAGE NO.

T01 tool no. 1 (.250 diameter, 2 flute end mill)

④ N20 S2000 M03

S2000 spindle speed set at 20000 rpm
M03 spindle on clockwise

⑤ N25 G00 X0 Y0 Z.100

G00 rapid traverse rate to X0Y0 at the top left corner of the part

Z.100 tool rapids down to within .100 of work surface

II hole drilling

① N65 G00 X.875 Y-750

Tool rapids to the top left hole location

② N70 G01 Z-.250 F10

Tool feeds .250 into work at 10 in/min to drill first hole

③ N75 G00 Z.100

Tool rapids out of hole to .100 above work surface

④ N80 X1.250 Y-1.125

Tool rapids to second hole location

⑤ N85 G01 Z-.250 F10

Tool feeds .250 into work at 10 in/min to drill the second hole.

⑥ N90 G00 Z.100

Tool rapidly out of hole to .100 above work surface.

III Turning Programming

Program sequence

% (reserved stop code / parity check)

2001 (Program number)

① N05 G20 G90 G40

G20 inch data input

G90 absolute positioning mode

G40取消 tool radius compensation

② N10 G95 G96 S2000 M03

G95 feed rate per revolution

G96 constant feed rate

S2000 spindle speed set at 2000 r/min

M03 spindle ON clockwise

③ N15 T02 02

Tool number & offsets

④ N20 G00 X1.200 Z.100

G00 rapid traverse mode

X + Z tool reference or change point

X1.200 tool point 100 away from outside diameter