



Application Program

IBM 1620 Drafting System

Application Description

The IBM 1620 Drafting System is a series of programs operating under monitor control to produce detailed engineering drawings on the IBM 1627 Plotter from English-like statements. IBM 1311 Disk Files are used to store data and programs. The input language consists of geometric, arithmetic, dimension, control and annotation statements.

INTRODUCTION

A large portion of the paperwork in the engineering profession consists of sketches, layouts, schematics, graphs, and other forms of engineering drawings. The IBM 1620 Drafting System — a series of programs using the power of the IBM 1620 Data Processing System to produce engineering drawings — provides a new tool for designers, engineers, and draftsmen.

In using the 1620 Drafting System, a series of statements is prepared using an English-like vocabulary to describe the drawing to be produced. The statements include geometric, arithmetic, annotative, and control information. Punched cards, containing the statements, are read by the IBM 1620, which performs necessary calculations, converts the statements into plotting instructions, and produces the finished drawing on the attached 1627 Plotter. Figure 1 shows a general flow diagram from the preliminary sketch to the completed engineering drawing.

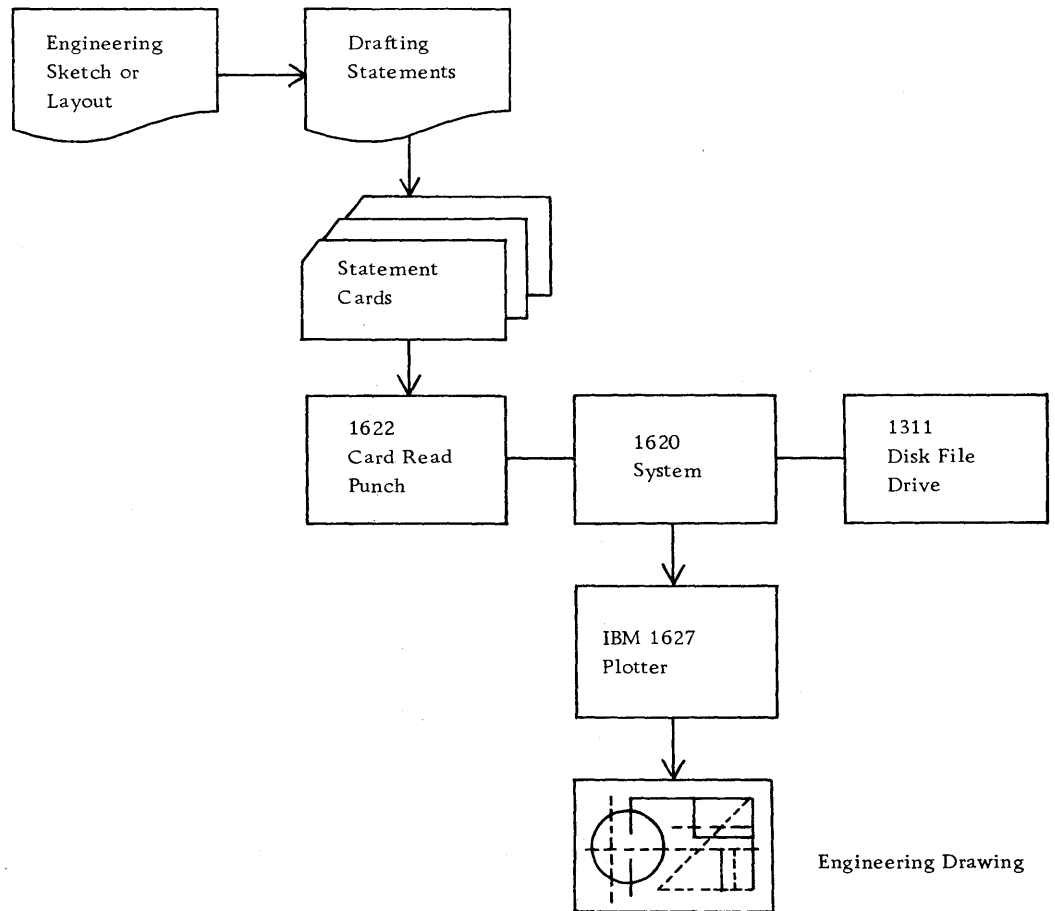


Figure 1. 1620 Drafting System

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N. Y. 10601

Features

- The first IBM system designed to produce detailed engineering drawings as computer output.
- One language with the power and versatility to do detail drawings and repetitive drawings.
- Language statements to drawing in one monitored job.
- IBM 1311 disk files used to store data and programs for rapid processing.
- User can store subdrawings on the file to suit individual requirements.
- An important step in an Engineering Information System.

INPUT LANGUAGE

There are basically four families of statements: geometric, arithmetic, annotative, and control.

Geometric Statements

These statements define the geometry of the part. Definitions may be written in terms of absolute coordinates, relative coordinates, or as functions of other geometric elements.

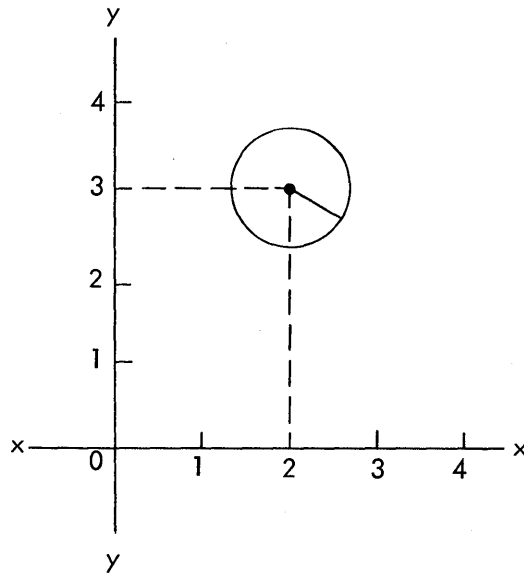


Figure 2.

The circle in Figure 2, for example, is defined in relation to the xy coordinate system. A statement defining the circle may be expressed as:

$$\text{CIRA} = \text{CIRCLE}/2, 3, 1$$

where CIRA is an arbitrary name for the circle, CIRCLE indicates the type of statement as circle definition, and the three digits represent the x coordinate, y coordinate, and radius in inches.

The center point of the circle in Figure 2 may be defined in a statement as:

$$\text{PTCTR} = \text{POINT}/2, 3$$

where PTCTR is an arbitrary name for the center point, POINT indicates the statement type, and the two digits represent the x and y coordinates.

Using the preceding statement, the circle in Figure 2 can now be defined as:

$$\text{CIRA} = \text{CIRCLE}/\text{PTCTR}, 1$$

since the symbol PTCTR has been previously defined by actual x and y coordinates.

Arithmetic Statements

In describing a part or drawing, arithmetic statements allow the definition of data which is not directly available. An arithmetic statement may appear by itself:

$$A = B + C/2.3$$

or in a nested statement such as:

$$\text{CIRCLE}/\text{PCTR}, (\text{DIA}/2)$$

The form of the arithmetic expression (AE) is:

$$L1 \text{ OP } L2 \text{ OP } L3 \dots$$

where OP is either add (+), subtract (-), multiply (*), or divide (/), and L is a label (such as PCTR), a number, another arithmetic expression (AE) enclosed in parentheses, or a function.

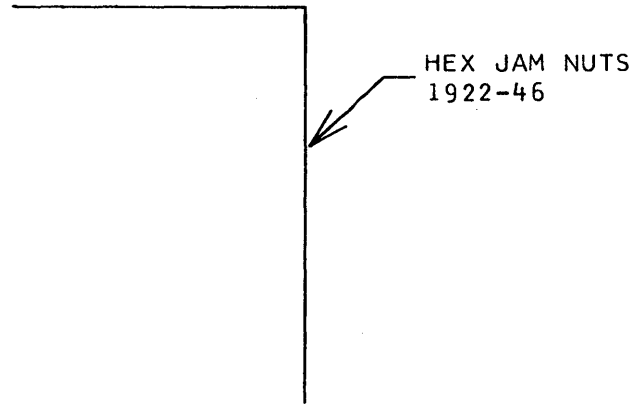
FUNCTIONS

B = SIND (AE)	Sine is saved in B	AE is value in degrees
B = COSD (AE)	Cosine is saved in B	AE is value in degrees
B = ATAND (AE)	Angle in degrees is saved in B	AE is value of tangent
B = DIST (P1, P2)	DIST between two points is stored in B	P1 and P2 are points
B = SQRT (A)	The square root of A is stored in B	A is a scalar

Annotative Statements

NOTE STATEMENTS

A sample of a note on a drawing and the necessary input statement is shown in Figure 3.



```
NOTE/P1,P2,@HEX JAM NUTS@,@1922-46@
```

Figure 3.

The following information is required to draw a note:

- A. The "from" point is the point on the body of the object to which an arrowhead points. This can be defined in any one of the point formats.
- B. The "to" point is the point where the textual information is placed.
- C. If only one point is present, it describes the beginning of the text.
- D. The text itself is a line of text written as it is to appear on the drawing with the addition of @ symbols at the beginning and end of the text.

Special drawing characters that are not available on a card punch, such as the symbol for center-line, may be expressed as:

\$ CL \$

where the CL is a mnemonic for center-line and the two dollar signs indicate that the mnemonic represents a single drawn character.

DIMENSION STATEMENTS

The statement in Figure 4 will, in general, generate two extension lines to the labeled point or line, put an arrow on each end of the line, and automatically compute the dimension.

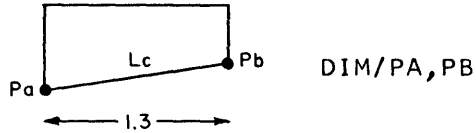


Figure 4.

In order to provide various formats, a "start dimension" statement is used to orient the dimension line, as illustrated in Figure 5.

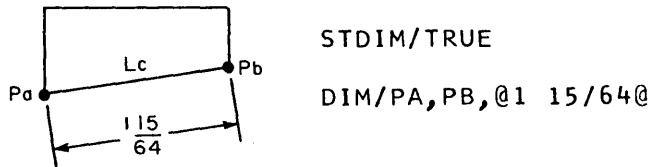


Figure 5.

Dimension notation uses the format:

DIM/PT (or line), PT (or line), @ (text) @

This statement is used for notes that are between extension lines. For example: the note INDUCTION HARDENING at the end of bar stock often has extension lines, dimension lines, and arrows, as shown in Figure 6. Any alphameric message or dimensions not requiring calculation may be specified.

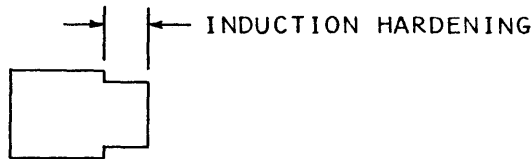


Figure 6.

Control Statements

Several control statements are available to control logic ability, program loops, macros, and line class capabilities. Two examples of control statements are the macro subpicture and the IF statement.

MACRO SUBPICTURE

The macro capability of the 1620 Drafting System greatly facilitates its use by the draftsman. Predefined subdrawings placed on the direct-access file are readily available to the draftsman. Elements that fall into this category are keyways, bolts, gear teeth, or any other element that occurs frequently and has a consistent geometric and arithmetic relationship. It is possible for whole drawings to be a macro or subpicture

when parts fall into part families. A slot would have as parameters length and width. By changing part family names it could be drawn as a rectangle, or with rounded corners, or the rounded corner could be an input parameter to a single macro. An example of a macro is the cap screw shown in Figure 7, which has been stored on the file and can be called by its assigned name and parameter list, using a statement such as:

CALL/CAPSCR, A = 2, B = .293, C = 7/16, D = 13/16

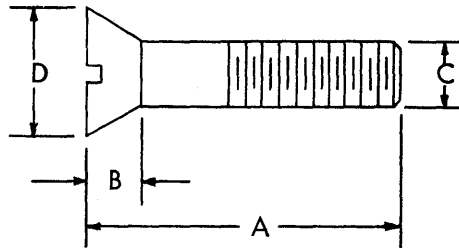


Figure 7.

The macro has returned many graphic elements and computed values that did not appear in the input statement. This provides a new powerful tool for the draftsman.

IF

During a program it may be desirable to make a decision at object time which will affect the sequence of executed statements. The standard FORTRAN "IF" statement is used for this purpose as follows:

IF (AE) S1, S2, S3

AE is any arithmetic expression. If AE is negative, statement S1 will be executed. If AE is equal to zero, statement S2 will be executed. If AE is positive, statement S3 will be executed. For example:

<u>Statements</u>	<u>Description</u>
A = 1	Define A
B = 1	Define B
1) LINE/P1, A, B	Draw a line from point 1 (P1) to the point defined by the x coordinate (A) and the y coordinate (B).
A = A + 1	Increment the x coordinate by 1.
B = B + 2	Increment the y coordinate by 2.
IF (B - 20) 1, 1, 2	If the arithmetic expression (B - 20) is negative or equal to zero, go to statement 1. If the AE is positive, go to statement 2.
2) POINT/	

GENERATED OUTPUT

The IBM Drafting System converts the statements that describe the drawing into commands which operate the IBM 1627 Plotter, producing a detailed engineering drawing. A sample drawing is shown in Figure 8.

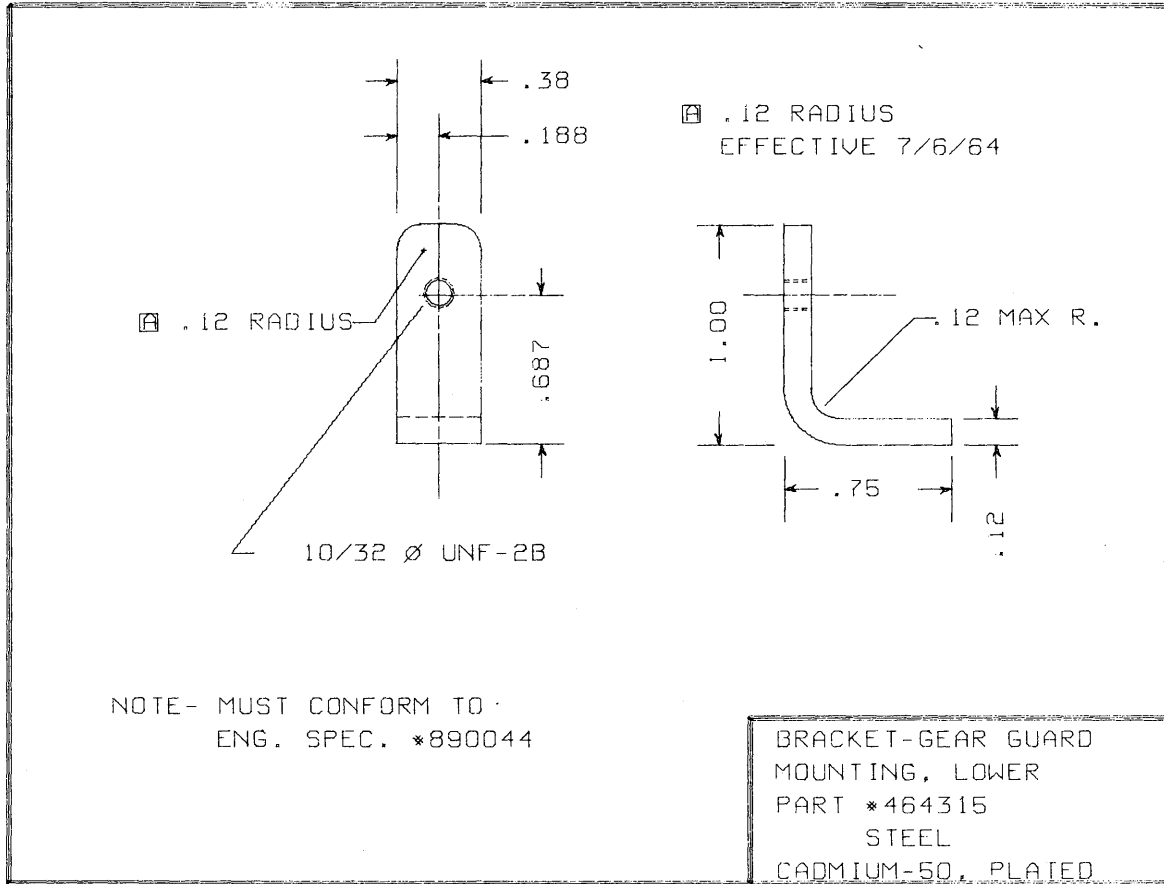


Figure 8.

MINIMUM SYSTEMS CONFIGURATIONS

IBM 1620 Data Processing System, 20K, Model 2
1622 Card Read Punch and Card Read Punch Adapter
1311 Disk Storage Drive, Model 3, and Disk Storage Drive Adapter
Paper Tape Adapter
1627 Plotter with Plotter Control

Or:

IBM 1620 Data Processing System, 20K, Model 1
Additional Instructions (TNS, TNF, MF)
Automatic Divide
Indirect Addressing
1622 Card Read Punch and Card Read Punch Adapter
1311 Disk Storage Drive, Model 3, and Disk Storage Drive Adapter
Paper Tape Adapter
1627 Plotter with Plotter Control

PROGRAM LANGUAGE

The IBM 1620 Drafting System is written in SPS II-D using Monitor control.



International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York 10601