# Computer aided Design and Fabrication of a Progressive Dieset for Thrust Washers

Project Report 1992-93

SUBMITTED BY

Jagadheswaran. V A

Ravishankar. N

Suresh. R

UNDER THE GUIDANCE OF

R. Gunasekaran M.E.

In Partial Fulfilment of the Requirements
for the award of the degree of
BACHELOR OF ENGINERING IN MECHANICAL ENGIEERING
of Bharathiar University

Department of Mechanical Engineering

Kumaraguru College of Engineering

COIMBATORE - 641 006

DEPARTMENT OF MECHANICAL ENGINEERING

# Kumaraguru College of Technology

COIMBATORE - 641 006

## CERTIFICATE

This is to certify that the contents of the Project Report entitled

**Computer aided Design and Fabrication of Progressive Dieset for Thrust Washers**

has been submitted by

Mr. N. RAVISHANKAR R. SURESH

In Partial Fulfilment for the award of

**BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING**

branch of the Bharathiar University, Coimbatore

During the academic year 1992-93

R. Gunasekaran
Guide

Head of the Department

Certified that the candidate was examined by us in the project work viva-voce examination held on .................................... and the University register number was ....................................

Internal Examiner

External Examiner

CONTENTS

1) Synopsis

2) Area & Application of Thrust washer

3) Problem identification and solution

a) Cost reduction

b) Process Improvement

4) Terminologies

a) Press working terminology

b) Progressive Die

c) Stripper & Stock guide

d) Strip feeding

5) Design procedure

6) Fabrication

7) Assembly

8) About ACAD

9) About LISP

10) Software

11) About Program

12) Using the program

a) User's guide

b) Program limitations

13) Inputs and Results

a) Inputs for the washer design

b) Outputs in the design & drawing

14) Conclusion

15) References

16) ACAD command list

17) LISP command list

# ACKNOWLEDGEMENT

We are very grateful to our Principal **Major T.S.RAMAMURTHY,** Kumaraguru College of Technology for his kind patronage.

We express our profound gratitude to **Mr.KANDASAMY,** Professor and Head of the Department for motivating us to do this project.

We sincerely thank our guide **Mr.R.GUNASEKARAN** for his guidance to do thie project.

Our heartful thanks to **Mr.V.GUNARAJ,** for his valuable suggestions and needful advice to coordinate this project.

We are very thankful to the faculty of Computer of science department particularly to **Mr.MANIKANDAN** and **Mr.SHANMUGA SHANKAR** for their guidance to develop this software.

We are very much thankful to **Mr.KUMAR NARAYANAN** Works Manager and **Mr.MAHALINGAM** Design Engineer **M/S BIMETAL BEARING LTD.,** for their valuable suggestions in the due course of the project.

Finally we like to thank all faculty members and friends for their kind help to go on with this project.

# SYNOPSIS

Our project work " **COMPUTER AIDED DESIGN AND FABRICATION OF A PROGRESSIVE DIESET FOR THRUST WASHERS** " is done by computer aided design and drawing. According to the washer dimensions the operations to be performed on the strip , the design is made. For the design and drawing the software is developed in LISP.

The main body of the software has four modules

a) Creation of different washer types

b) Design of Punch and Die plate

c) Drawings of Punch and Die plate

d) Assembly Drawing

The first module computer selects the type of washer the designer need and gets the dimensions of the particular washer and washer drawing is made.

In second module, computer outputs the minimum values of different components with which the design is safe by getting the value of strip thickness.

In third module, based on the previous washer dimensions the punch and die drawings are drawn.

In fourth module, the die and punch drawings are be inserted in the assembly drawing to get the complete assembly of the thrust washer. This results obtained from the computer is used for fabrication purpose.

To assure a generalised program for different contour of Thrust washer seperate programs for those contours have been developed and hence the program.

# AREA & APPLICATION OF THRUST WASHER

This Thrust washer is invariably used in all engines for a specific purpose. While starting the engine there is an axial thrust developed which reacts to move the crank in the axial direction which goes and rubs against the crank casing. Thereby wears out the crankcasing. If this continuos the casing has to be replaced which proves to be a costlier one. In order to overcome this problem the thrust washer has been introduced in the crank casing and two sets of thrust washer on either side of the crank is fitted in the casing. During overhauling these thrust washers can be replaced with ease and further more which is cheap compared to crank casing. This thrust washer uses soft material compared to the that of crank and crank casing so that wear and tear on the part of the crank and crank casing is greatly reduced. This usage thereby increases the life of the engine particularly to the crank assembly.

This thrust washer has to be developed with greater accuracy so that proper setting in the crank casing can be ensured. Further more, provisions should be given for the lubricant to circulate and the washer should be designed in such a way that washer doesn't slips as the engine starts running.

# PROBLEM IDENTIFICATION AND SOLUTION

Various problems are being faced with the existing way of manufacturing and design of the dieset. Among this problems which has to be attended immediately are

a) Higher cost for fabrication of diesets for different washers

b) Wastage of Material

c) Wastage of resources

d) More downtime of the Machine

Higher cost of diesets is due to the problems in existing design, since different operations for the thrust washer is being performed in different presses. For that different diesets are to be manufactured for different contours of thrust washers and for different dimensions of that. Wastage of material is incurred because of the setting problem. Everytime if you change the setting , you are going to get defective pieces unless you setting is going to be correct and is according to specifications desired. Further more, this loss is incurred in both the presses.

Wastage of man power, is in the sense that the operations are being performed in different presses , for each machine an operator has to be engaged and excessive wastage of man power during setting. Excessive downtime of the machine is caused due to the excessive setting time and synchornisation between the two operations has to be achieved.

## COST REDUCTION

At present using a simple die, two different dieset each for one for Blanking and Embossing operation has to be fabricated. This dieset design has to be changed as and when Thrust washers of different thickness has to manufactured. Moreover, for different contours of thrust washer different dieset suitable for that contour and dimension has to be made.

Following the existing pattern means loss to the company since it reduces the margin of profit. In addition this pattern utilises separate man power and a separate presses for Blanking and embossing operations put together takes away more money and man power. A remedy to this problem is our new design,

## A PROGRESSIVE DIESET,

which uses a single press and can acco-modate the die and punch of different contours of thrust washers which thereby reduces the setting time and wastage of material which thereby reduces manufacturing cost.

## PROCESSING IMPROVEMENT

In the existing design, first Blanking operation is carried out and then the blanking operation in separate presses. Therefore more pieces are found to be defective due to setting problem in the second operation since synchornisation can't be achieved so easily as desired.

Improving the process also means increasing the accuracy of the thrust washers so that interchangeblity can be easily achieved in the crank assembly. This also reduces the cost of inspection and manufacturing cost.

## PRESS WORKING TERMINOLOGY

In dieset certain Press working terminologies are used which are defined below.

### BED:

The bed is the lower part of a press frame that serves as a table to which a bolster plate is mounted.

### BOLSTER PLATE:

This is a thick plate secured to the press bed which is used for locating and supporting the die assembly. It is usually 5 to 12.5 cm thick.

### DIESET:

It is unit assembly which incorporates a lower end upper shoe, two or more guideposts and guidepost bushings.

### DIE:

The die may be defined as the female part of a complete tool for producing work in a press. It is also referred to a complete tool consisting of a pair of mating members for producing work in a press.

**DIE BLOCK:**

It is a block or a plate which contains a die cavity and is fastened to the lower shoe by fasteners.

**LOWER SHOE:**

The lower shoe of a dieset is generallu mounted on the bolster plate of a press. The die block is mounted on the lower shoe. Also, the guide posts are mounted on it.

**PUNCH:**

This is the male component of the die assembly, which is directly or indirectly moved by and fastened to the press ram or slide.

**UPPER SHOE:**

This is the upper part of the dieset which contains guidepost bushings.

**STRIPPER :**

It is a plate which is used to strip the metal strip from a cutting or noncutting punch or die. It may also guide the sheet.

## SHUT HEIGHT :

It is the distance from top of the bed to the bottom of the slide, with its stroke down and adjustment up.

## PROGRESSIVE DIE :

A progressive or follow-on die has a series of stations. At each station, an operation is performed on a workpiece during a stroke of the press. Between stroke, the piece in the metal strip is transferred to the next station. A finished workpiece is made at each stroke of the press. While the stamp enbosses in the stock, the blanking punch blanks out a portion of metal in which embossing is done at the previous station. Thus after the 1st stroke only embossing will be done. Each stroke of the press produces a finished washer.

At the end of the stroke the stock is moved by the pitch distance of the washer by a feeder. The accuracy of the washer depends upon the feeder accuracy. So the accurately set for the pitch distance of the thrust washer.

## DROP THROUGH THE DIE :

In this dieset the dieblock assembly is mounted on the bolster plate and the punch assembly on the ram. The blank drops out of its own weight through the die opening provided and the clearance hole provided in the bolster plate and press bed. This type of drop through die is followed since it is economical and fast in working.

If in the washer an oilgroove has to be provided then drop through die cannot be adopted since during blanking the oilgroove made previously in the strip get spoiled. So we have to go in for inverted type die. In this type the removal of blank from the die is done by a device to knock the blank out of the die opening.

**Stripper**

After a blank has been cut by the punch on its downward stroke, the scrap strip has the tendency to expand. On the return stroke of the punch, the scrap strip has the tendency to adhere to the punch and be lifted by it. This action interferes with the feeding of the stock through the die and some device must be used to strip the scrap material from the punch as it clear up the die block. Such a device is called "stripper" **or stripper plate. Strippers are of two** types Fixed or stationary and spring loaded or movable. As it is clear, it is a plate parallel with and above die surface. An opening is cut through the stripper plate for free passage of the punch. By rule of thumb this opening can be about 1.6mm larger than the blank size on all sides.

## Fixed Stripper

This stripper is attached at a fixed height over the die block. This height should be sufficient to permit the sheet metal to be fed freely between the upper die surface and the under surface of the stripper plate. The stripper plate is usually of the same width and length as the die block. In simple dies, it is fastened with the same screws and dowels which are used for die block. In complex dies, the stripper fastening will be independent of die fastening. The thickness of the strpper plate should be sufficient to withstand the for ce needed to strip the scrap strip from the punch. the usual value is 9.5 to 16 mm. The following emprical formula may also be used for determining stripper plate thickness,

$$ts = 1/8 [w/3 + (16 * t)]$$

where w and t are the width and thicknes of stock strip.

The thickness of the stripper plate may also be determined by the size of the socket-head cap screws used to hold it in place. The stripper plate must be thick enough to allow for the screw-head counter bores, which in most cases provided adequate stripper strength. In addition to screws, dowels are used to ensure accurate alignment of the die block. Stripper may be made of Mild Steel. The fixed stripper is also known as **"channel stripper"**.

For the stripping action, on the upward movement of the punch, the scrap strip will strike the underside of the stripper plate and get stripped off from the punch. The underside of the stripper plate which comes in contact with the strip should be machined nad preferably ground. The height of the stock strip channel should be at least equal to 1.5 times the stock thickness. If the scrap trip is to be lifted over a fixed pin stop, this height should be increased. The width of the channel should be equal to the width of the stock plus adequate clearance. The disadvantages of fixed stripper are that it hides the work from the operator and it would interfere with removal of the scrap strip in large blanking operations

## STOCK GUIDE

Stock guide is the space provided in the dieblock, through which the stock strip is guided as it is fed into the die. The design of the stock guide will depend upon the type of stripper. For fixed strippers (channel strippers) the stripper acts as a pressure pad and presses directly against the stock strip. The stripper does not contact the guide rails.

## STRIP FEEDING

Stock strip may be fed into the die either manually or mechanically (automatic feeding) Manual feeding is suitable only for low production or with presses operating at low values of strokes per min. Modern presses operate at 200 to 300 strokes per min (may be up to 500 strokes per min.). For such cases,manual feeding is not feasible and automatic feeding is the only answer. For this,the strip is prepared inlarge coils. two methods are in use for this purpose.

(a) Reel

(b) Coil cradle.

The reel is considered to be better as it does not damage the strip in any way. The reel may be or may not be power driven. In the case of power driven reels, a roller at the end of a long loop arm, rides on the un coiling strip. When sufficient coil has been unwound, the loop arm actuates a switch which stops the power drive. Now, as the strip is used up and the loop arm gets raised,the power supply is switched on. In the case of unpowered reels, the coil is unwound by an external power source, which may be feeding mechanism or straightening rolls. When enough coil has been unwound the reel is stopped from uncoiling by a manual or automatic brake.

In the case of cradle, the strip is supported on the outside diameter of the coil. The coil locates against the rollers and as a result of this scratches may form on the coil.

The second step in stock feeding is straightening of the uncoiled strip. This is done to remove wrinkles and curvature from the strip.For straightening the coil, it is passed through inbetween a series of rollers.

After uncoiling and straightening, the final step is to feed the strip into the die. The two main types of feeding system are:

1. Roll feed
2. Slide or hitch feed

The roll feed moves the stock between a pair of rollers, which is driven through an overrunning clutch or ratchet mechanism timed from the press main shaft or ram. Roll feeds may be : single or double. In the case of single roll feed the rollers are provided only on one side of the press and they push or pull the strip through the die. In the case of double roll feed,feeding mechanism is mounted on each side of press bed with a drive connection between them. One feed pushes the stock and the other pulls stock through the die. This will keep strip tight and prevent its buckling.

In the case of slide or latch feed , grippers are used which grasp strip mechanically and feed it into the die by a reciprocating mechanism which may be driven from the press crankshaft, a cam mounted on the punch holder, hydraulically or pneumatically.

# DESIGN PROCEDURE

The design procedure is to make out the thrust washer with the existing facilities and for the different dimensions of the component. With that in mind taking out the maximum dimensions of thickness, radius and press tonnage the design is carried out. We have made maximum usage of existing facilities.

This design procedure is the minimum values of die and punch so that even the maximum component dimensions can be manufactured. Although, proposed values may be more than the design values so that the manufacture of different thrust washers can be made possible.

## PROCEDURE

Thickness of the workpiece material ( Maximum thickness of the workpiece) = 4.25 MM

Cutting perimeter =

(2 * 3.14 * outer radius) + (2 * 3.14 * inner radius) + 2 * (outer radius - inner radius)

Maximum outer radius = 63.5 MM (2.5")

Inner radius = 50.8 MM (2")

Therefore,

Perimeter = 398.982 + 319.186 + 25.4

= 743.568 mm

Therefore,

Dieplate thickness = 14 mm

Margin on the site of Die opening

= 3 * Dieplate thickness

= 3 * 14

= 42 mm

Add grinding allowance = 5mm to dieplate thickness

Therefore,

Die plate thickness = 19mm

Production qty (million) = Unlimited

For stock thickness of 4.25 mm Die block thickness should be 37.5 mm

Area of the die block = width * length

Width = outer radius + 2 * margin

= 147.5 mm

Length = [2 * outer radius] + [2 * margin]

= 211 mm

Area of the die cavity = 31122.5 Sq mm

Recommended sizes of screw for fixing the die with dieset

Size of screws = M 12

Number of screws = 4

Selection of die pillar set with rectangular working area.

A four pillar dieset of working area [ 211 * 147.5 ] is selected

Working surface = 315 * 160

Length of dieset = 450 mm

Breadth of dieset = 340 mm

Pillar diameters = 30 & 32 mm Thickness of lower shoe = 50 mm

Thickness of upper shoe = 65 mm

**CENTRE OF GRAVITY**

Outer area = 3.14 / 2 * (Outer radius) 2

Inner area = 3.14 / 2 * (inner radius) 2

Total area =

3.14 / 2 * [(Outer radius) 2 - (Inner radius) 2]

= 3.14 / 2 * [(63.5) 2 - (50.8) 2)]

= 2280.1837 Sq mm Area * Y

= 3.14 / 2 *

[ ( (Outer radius ) 2 * 4 * outer radius / 3 * 3.14 ) - ((Inner radius ) 2 * 4 * inner radius / 3 * 3.14))] * 2/3*((outer radius) 3- (inner radius) 3 ]

Therefore,

Y = 4 / ( 3 * 3.14 )

( (outer radius ) 3 - ( inner radius ) 3 ) /

( (Outer radius ) 2 - ( inner radius ) 2 )

= 36.53 mm

Since the washer is symmetrical about Y-axis the centre of gravity is 36.53 mm from the centre point of the washer in the Y-axis.

**Percentage of penetration** = 56% of stock thickness

= 0.56 * 4.25

Punch penetration = 2.38 MM

## PRESS TONNAGE

Press tonnage = Shear stress * Perimeter *

thickness = 385 expt 10 -2 N/Sqmm

* 743.568 * 4.25

= 121.66 Tonnes.

Length of the punch = ( (3.14 * d ) / 8 ) /

[sqrt ( E * d / Shear

stress*t)]

Punch and die material = High Carbon High chromium

steel

Cold drawn and temper = 440 N / Sq mm

E = 2.08 * 10 5 N / Sq mm

= ( 746.368 / 8) /

[sqrt (2.08 * 10 5/ 440 *

4.25)]

Punch length = 19.86mm

## STRIPPER PLATE DESIGN

Stripper plate thickness = ( ( W/3 ) + 16 T ) * 1/8

W = Width of the stock

T = Thickness of the stock

Stripper plate thickness = [(140.81/3) + 16 * 4.25) * 1/8]

= 14.367 mm

Height of the stock strip channel

= 1.5 times stock thickness

= 1.5 * 4.25

= 6.375 mm

Width of stock strip channel

= Width of stock + clearance

= 127 + (2 * 6.35)

= 139.7 mm

## CLEARANCE

Clearance between punch and die (Note: In blanking operation, where the blank is to be held in size, die size should be of the same size of thrust washer and the punch size should be obtained by substracting the clearance from the die opening size.)

Clearance = 7 % of Thickness of strip

= 0.07 * 4.25

= 0.2975 mm

Therefore, clearance between punch and die on either side

= 0.35 / 2

= 0.175 mm .PA

This clearance is used for deriving the punch size by subtracting clearance from the die opening size.

## CALCULATION FOR PITCH DISTANCE

Pitch = Inner radius + b

For material thickness 4.25 mm

b = 3.2 mm

Therefore,

Pitch = 50.8 + 3.2

= 54 mm

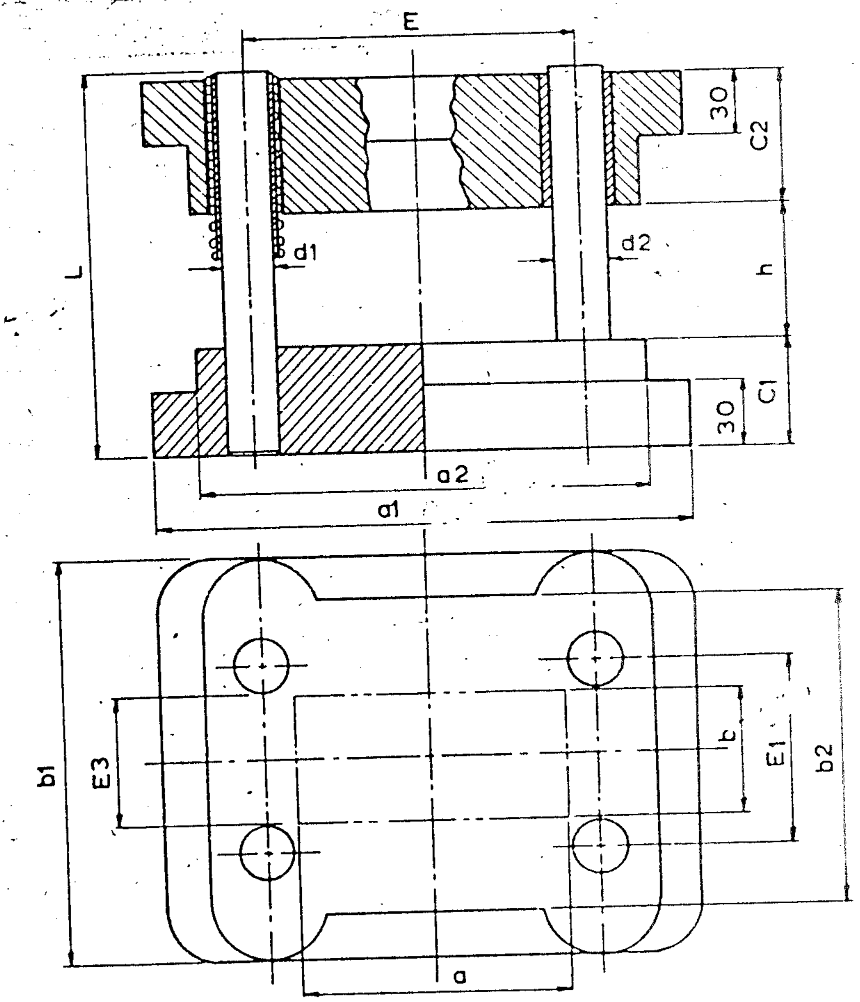
Therefore, distance between punch and stamp is multiple times of pitch, [ taking m = 3]

= M * Pitch

= 3 * 54

= 162 mm

## FOUR PILLAR DIE SET WITH REACTANGULAR WORKING AREA



All dimensions in mm

| Working surface a & b | a1 | a2 | b1 | b2 | E | d1 | d2 | E1 | E3 | C1 | C2 | h | L | Standard Threads |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 125 × 63 | 250 | 210 | 190 | 150 | 155 | 24 | 25 | 90 | 61.5 | 50 | 65 | 65 | 180 | M30 × 2 |
| 160 × 63 | 275 | 240 | 190 | 150 | 190 | 24 | 25 | 95 | 69.5 | 50 | 65 | 65 | 180 | M30 × 2 |
| 200 × 100 | 340 | 300 | 240 | 200 | 230 | 24 | 25 | 130 | 104.5 | 50 | 65 | 85 | 200 | M40 × 2 |
| 250 × 100 | 390 | 350 | 240 | 205 | 285 | 30 | 32 | 135 | 104 | 50 | 65 | 85 | 200 | M40 × 2 |
| 315 × 160 | 450 | 440 | 340 | 290 | 350 | 30 | 32 | 200 | 169 | 50 | 65 | 85 | 200 | M50 × 2 |
| 355 × 250 | 535 | 480 | 430 | 380 | 395 | 38 | 40 | 290 | 253 | 50 | 65 | 85 | 200 | M50 × 2 |
| 400 × 250 | 545 | 495 | 430 | 380 | 445 | 38 | 40 | 290 | 253 | 50 | 65 | 85 | 200 | M50 × 2 |

## RECOMMENDED DIE PLATE THICKNESS

(for dies with cutting perimeter less than 50 mm)

| Stock thickness, mm | 0.25 | 0.5 | 0.75 | 1 | 1.25 | 1.5 | 1.75 | 2 | 2.25 | 2.5 |
|---|---|---|---|---|---|---|---|---|---|---|
| * Die plate thickness, mm/shear stress kg/mm² of stock | 1 | 2 | 3 | 4 | 4.5 | 5.25 | 5.75 | 6.3 | 6.7 | 7 |

* If the die is located on a shoe, the thickness can be decreased upto 50%. Grinding allowance of 2.5 to 5 mm to be added to the thickness calculated. Minimum thickness of die plate shall be 10 mm.

## DIE PLATE THICKNESS

(for dies with cutting perimeter greater than 50 mm)

| Cutting perimeter, mm | | | | | |
|---|---|---|---|---|---|
| | over | 50 | 75 | 150 | 300 |
| | upto | 75 | 150 | 300 | 500 |
| factor by which the above tabulated values should be multiplied | | 1.25 | 1.5 | 1.75 | 2.0 |

## MARGIN ON THE SIDES OF DIE OPENING

For small dies (1.5 to 2) × Die thickness

for large dies (2 to 3) × Die thickness

## MINIMUM CRITICAL AREA vs IMPACT LOAD

| Impact load, tonne | 20 | 50 | 75 | 100 |
|---|---|---|---|---|
| Critical area = margin × thickness of die, sq. mm. | 315 | 625 | 1000 | 1250 |

## RECOMMENDED SIZES OF SOCKET HEAD CAP SCREWS FOR FIXING THE DIE

| area of die block, mm² | over<br>upto | <br>4500 | 4500<br>6300 | 6300<br> |
|---|---|---|---|---|
| * Size of screws | | M 10 | M 10 | M 12 |
| No. of screws | | 2 | 3 | 4 |
| Size of dowel, mm | | 10 | 10 | 12 |
| No. of dowels | | 2 | 2 | 2 |

* Counterbore depth for screws should be deeper by grinding allowance.

## FABRICATION

This design is carried out with the view of sorting out the problems mentioned previously. Versatality of the design is achieved with the following way of design with this design accuracy of the thrust washer is very much enhanced and thereby cost incurred for manufacturing, inspection, materials are lowered.

### DIESET

The design for the dieset is forwarded taking in the minimum values calculated previously for the maximum dimensions of the thrust washer. In this a four pillar dieset is selected considering the area of the die cavity. The thickness of the dieset is taken, for the maximum thickness of the washer ie., 4.25 mm so that it can withstand the cutting forces exerted during the operations. The pillars are set in the dieset so that upper shoe is guided properly with respect to lower shoe.

For the lower shoe a cast block of 650 mm x 550 mm x 65 mm is taken. The thickness of the block is reduced to 62.55 mm by milling using a milling machine. The width and length is reduced to 546 x 635 mm by milling.

A gateway of 635 x 304.8 mm is taken by milling operation which is placed symmetrically about X-axis. On the bottom side two holes for a depth of 12.7 mm and a diameter of 101.6 mm with four centre to centre distance of 495.3 mm in the X-axis. Similarly another set of holes is drilled with the same dimensions with centre to centre distance of 40.4 mm in the Y-axis using a radial drilling machine.

A keyway of 38.1 x 546 mm with a depth of 12.7 mm is taken at a distance of 194.85 mm to the left hand side of RAM axis. Similarly another key way is taken at a distance of 82.55 mm placed at the right hand side of the RAM axis. A threaded hole of 13/32'' six in number using milling operation is done first by drilling and then by tapping. Three holes are placed on either side with respect to the centre axis with centre to centre distance of 241.3 mm. Out of the three holes the first hole is at a distance of 38.1 mm from the right keyway. The second hole is placed with the centre to centre distance of 47.625 mm from the second hole. Similar procedure is carried out for the upper shoe with similar dimensions.

In the lower shoe a cavity of 88.9 mm x 152.4 mm is taken out by milling operation. A cavity is placed, 38.1 mm in the right side in the X-axis from the RAM axis.

## PILLAR

This pillar is manufactured separately and is inserted in the dieset. A shaft of diameter 105 mm and length 305 mm is taken and is held in the lathe. It is turned to a diameter of 101.6 mm through the whole length. Then facing is done in the shaft on either side, the length is reduced to 304.6mm using a lathe. Then in the same setting turning is done and the diameter is reduced to 76.2mm at a distance of 12.7 mm from one end for a length of 279.2 mm. Totally, four pillars are required with similar dimensions for guiding the upper block with respect to the lower block.

The material used for manufacturing the dieset is high carbon high chromium steel and note that hardening is not done for this dieset.

**DIE**

A cast block of 180 mm x 220 mm with height of 55 mm made of high carbon high chromium steel is taken using end milling operation this is reduced to 177.8 x 215.9 mm and height is reduced to 50.8 mm. A slot of 177.8 x 25 mm and a height of 30.16 mm using end milling operation. The ram axis passes through the block at a distance of 82.55 mm from the right side. The intersection of the Ram axis, centre axis in the x-direction should be the centre of gravity, of the die cavity.

The die cavity is taken by wire cutting exactly to the same size of the thrust washer required. Fine grinding is done in the inner side of the cavity. This cavity is taken out for the whole height of block 50.8 mm.

Hardening is done by the hardening procedure for a Rockwell hardeness of RC 60.

## STRIPPER PLATE

Cast block of material, mild steel of dimension 180 x 195 mm with a height of 20 mm is taken out. A slot of 165.1 x 190.5 mm with a height of 6.35 mm is taken out at a distance of 6.35 mm is taken out at a distance of 6.35mm from the right end. After this operation the piece is reversed and four threaded holes are drilled and then tapped. Two are placed on one side, two on other side with respect to the centre axis in the x-direction with centre to centre distance of 158.75 mm at a distance of 19.05mm from the right end of the block in the x-axis.

A cavity whose contour is similar to that of the thrust washers and size increased by the clearance of 5mm is taken out using end mill cutter and finished by grinding. The countour and the intersection of the ram axis and centre axis in the x-direction. The ram axis is placed at a distance of 88.9 mm from the right end of the block.

## PUNCH

A cast block of material high carbon high chromium

## HARDENED PLATE

A cast block 280mm x 220mm with the height of 20mm of material High carbon, High chromium steel is taken and using end milling the dimensions are reduced to 279.4mm x 215.9mm the height is reduced to 19.05mm. Six holes are drilled using Radial drilling machine, the drilling operation is carried out. The holes are then tapped. First the holes are drilled for a diameter of 13/32" and then enlarged to a diameter of 19/32" for a depth of 15.875mm from the top of the block. Three holes are placed on one side and other three on the other side. The holes are placed apart with a centre to centre distance of 241.3mm from the centre axis in the x-direction. The first hole is placed 38.1mm from the end of the block and next hole 66.675mm centre to centre and the third hole 47.625mm centre to centre from the second hole.

Similarly another block is done with similar dimensions and in this block a cavity of dimension 88.9mm x 152.4mm. This cavity is placed at a distance of 38.1mm on the right hand side of the ram axis. The ram axis is at a distance of 82.55mm from the right end of the block. This cavity is also taken by end milling operation.

Hardening is done using hardening procedure for a Rockwell hardness of RC60.

## SETTING BLOCK

Material used for this guide block is EN8 and hardened using hardening process for a rockwell hardness of RC35.

A cast block of dimension 45mm x 550mm and to a height of 35mm and this is reduced to 44.45mm x 546mm and the height to 30.16mm using end milling.

A similar block is fabricated with similar dimension and a threaded hole of diameter 3/8" is done first by drilling and then by tapping. The threaded hole is placed symmetrically about x-axis and at a height of 19.05mm from the top of the plate.

Totally, four guide blocks, two guide blocks using first procedure and two blocks using another procedure is fabricated.

## HARDENED PLATE FOR STAMP HOLDER

The material used is High carbon High chromium tool steel, hardened using hardening procedure for Rockwell hardness of RC35.

A cast block of dimension 105mm x 220mm and with a height of 55mm is taken and the dimensions are reduced to 101.6mm x 215.9mm and a height of 50.8mm using end milling.

A slot is taken for dimension 101.6mm x 25.4mm and for a depth of 30.16mm placed at a distance of 95.25mm from the centre axis in the x-direction. Another slot is taken with similar dimensions but at the lower side. A hole of diameter 3/8" for a depth of 12.7mm placed in the x-axis at a distance of 88.9mm from the right end of the block.

Thereby the procedure is carried out for having a similar block with similar dimensions. And in this block four threaded holes are made, first by drilling and then by tapping for a diameter 3/8". Two holes are placed on one side and two on other side about the centre axis in the x-direction. The centre distance being 158.25mm. The first hole is placed at a distance of 12.7mm from the right end and the second hole is placed at a distance of 12.7mm from the left end . The holes are made for fixing the stamp holder.

## CLAMPING PLATE

Material used for the clamping plate is Mild Steel. A cast block of 45mm x 280mm and a height of 15mm and the dimensions are reduced to 44.45mm x 279.4mm and a height of 12.7mm. Using end milling a slot of dimensions 31.75mm x 279.4mm and for a height of 4.7mm is taken. Then the piece is reversed and six threaded holes are drilled and tapped for a diameter of 3/8". The first hole is placed at a distance of 19.05mm and second hole at 44.45mm centre to centre from the first hole. The third hole at 50.8mm and the fourth being at 44.45mm centre to centre from the third, firth hole at 38.1mm centre to centre from the fourth and finally, the last hole at 63.5mm centre to centre from fifth hole.

Totally, four clamping plates are fabricated using above procedure.

## GUIDE BLOCK

Material used for this guide block is Mild Steel. A cast block of 25mm x 195mm at a height of 7mm is taken out and milled to a dimension 22.15mm x 190.5mm and height being 6.35mm. Six holes are drilled and then tapped for a diameter of 3/8" placed symmetrically about y-axis. The holes are with the centre to centre distances of 19.05mm from one end, 44.45mm, 50.8mm, 44.45mm, 38.1mm, 63.5mm being the centre distances between successive holes.

Similarly, another block is done with similar dimensions. These blocks are placed on the top of the die plate on either side.

## STAMP HOLDER

The material of this stamp holder is High carbon High chromium tool steel. The outer dimensions of this being 50.8mm x 76.2mm and for this a cast block of 55mm x 80mm and milled out for required dimensions. A slot of 50.8mm x 12.7mm is milled and placed on either side at a distance of 25.4mm is taken out for fixing the monochrome in this slot.

## STAMP

In this, part number are scribed out on the block of dimension of 20mm x 7mm and height of 16mm. The stamp projection carrying the part number being 1mm. Then, the letters are on a seperate monochrome whose dimensions being 2mm x 16mm and height of 4mm.

These monochrome is placed on the side of the stamp bearing the part number.

## HARDENING PROCEDURE

High carbon High chromium tool steel is first air hardened and then stress relieved which possesses extremely high wear resisting and practically free from size change after proper treatment. This tool steel's high chromium content gives it mild corrosion resisting properties in the hardened condition. This tool steel after hardening is used for blanking dies and punches.

After hardening it is quenched in air at 1010 c and tempered for one hour and then stress relieved.

High carbon High chromium steels achieve excellent wear resistence due to chemical balance which renders them notch sensitive and low in ductility. Compressive loads of 400,000 psi (2758MPa) can be withstood if evenly applied at low rates of loading.

## ASSEMBLY

The fabricated components should be properly assembled with ease so that machine downtime is greatly reduced. Only once the dieset with some specific parts is assembled for a particular thrust washer. If some other washer has to be produced then only the die, punch, stripper plate and the monochrome and the stamp will be changed. With this setup the setting time is drastically reduced.

## ASSEMBLY

The dieset is placed on the bolster plate and with the help of four hexagonal bolts lower shoe is fastened to the bolster plate which in turn is fastened to the press. Then, the upper shoe, is fastened to the ram with the help of four hexagonal bolts. Setting block is placed on the right slot of the dieset. Hardened plate which is used to take the cutting forces as the dieset is not hardened without which damage is caused to the dieset during shearing.

The hardened plate is placed in such a way that the cavity provided is nearer to the setting block placed prior to this. Another setting block with provisions for nine threaded holes is placed on the upper side of the centre axis in the x-direction and is fastened with the help of socket screws to the dieset. Similar setting block is placed in the lower side of the centre axis and this is also fastened with the help of socket screws. Then, the dieblock is placed on the hardened plate wherein the cavity is through. With the help of clamping plates these are fastened to the dieset. The clamping plates are first fastened to the setting block with the help of socket screws. Altogether, eight socket screws are required for this purpose four on the upper and four on the lower. This clamping plate seats on the die slot which clamps the die.

The guide block which is used for guiding the stock is then placed over the dieplate and over which the stripper plate is placed and with the help of four socket screws it is tightened to the die.

The hardened plate for the stamp holder is placed on the left side of the die and is fastened together with the help of clamping plates by socket screws. Then, the setting block is placed and with the help of the socket screw the hardened plate and die is butt against the other setting block.

Similarly, for the upper shoe the above assembly procedure is done wherein the this won't have the guide block and stripper plate but others remaining the same.

The stamp holder is fixed in the hardened plate of the top shoe placed on the side of the punch collar with the help of four socket screws to the hardened plate. Then the monochrome is placed inside the cavity of the stamp holder and then the stamp bearing the part number of the particular thrust washer. With the help of sixed socket screws this monochrome and stamp is butt.

The stock is passed inbetween the rollers of the feeder and above the uppershoe and through the guide block.

The feed rate of the feeder is set according to the pitch distance of the washer. Note that the maximum shut height of the press is only 304.8mm (12").

Over prolonged use the edges of the punch and die may wear out and regrinding is done which inturn decreases the shut height which is slightly increased to optimal level with the provisions provided in the press.

# ABOUT AutoCAD

## INTRODUCTION

The AutoCAD drafting package is a general purpose Computer- Aided drafting application for the computer CAD applications are tremendously powerful tools. The speed and cas with which a drawing can be prepare and modified using a computer offers a phenomenal time saving advantage over "head" prepbration. AutoCAD brings this sophisticated technology, previosly available only on large and costly systems, to us, the desktop computer user. There is virtually no limits to the kinds of lines drawing we can prepare using AutoCAD. If a drawing created by hand, it can be generated by the AutoCAD. AutoCAD remembers the locations, sizes and colors of the objects we draw, maintaining them in a database for subsequent retreval, analysis and manipulation. The AutoCAD is available for a variety of desktop computers and engineering workstations, and runs under the PC-DOS, MS-DOS, UNIX, AEGIS, and UMS operating systems. No technical computer knowledge is required to aid AutoCAD effectively; practice and a thorough understanding of its features are the keys to proficiency.

## REQUIRED EQUIPMENT

In addition to a basic computer system (including processor, keybord, text display screen and drives) AutoCAD requires a graphic monitor capable of reasonably high resolution for computers base on INTEL 8086 family of Microprocessors, an 8087, 80287, or 80287 math coprocessor is also required.

## DISPLAY MONITOR

On some computers, AutoCAD uses two display monitors, one for command prompts and text output, and the other graphics on these systems, the graphics monitor can also display a screen menu along its right edge and a one line prompt area accross the bottoms, as in the following illustriations. The right edge can contain a screen menu when running on a single screen system, AutoCAD remembers a full 24 lines of text, just like the regular text display. If information two scrolled of the three line display, we can text display when drawing anything.

## PEN PLOTTERS AND PRINTER PLOTTERS

A pen plotter or a printer plotter can be connected to the system to produce a "hard copy" of a drawing. AutoCAD can support one plotter and one printer plotter on the same system. Some of these devices connect to an RS-232C serial communication parts. The models supported on each computer and instructions for their installation and use vary with each model of AutoCAD.

## POINTING DEVICES

A pointing devices such as mouse or a digitizing tablet provides the means for instant command and point entry. Keyboard entry is relatively easy, but pointing at the screen and pushing a button is even easier. In addition to locating points and entering commands, we can use a digitizing tablet to trace over existing drawings. Descriptions of the types of devices available appear below.

## MOUSE

As we move a mouse around the table top, crosshairs track Mouse Movement on the screen. To select a point or menu items position the crosshairs on it and push the button on the mouse. If the mouse has multiple buttons, they are used to invoke frequently-needed AutoCAD commands.

## TABLETS

Point and menu item selection using a digitizing tablet are similar to the mouse operation described above. However, we move the tablet's puck or stylus around only on the tablet's surface. The tablet offers two capabilities beyond those of the mouse. we can align the tablet with the coordinate system of an existing paper drawing so that we can use AutoCAD to trace over it. we can set aside upto four areas of the tablet for tablet menus.

## CONCEPTS AND TERMINOLOGY

This section presents some terms and concepts we will encounter while working with AutoCAD. These items are best understood by working with the program.

## AutoCAD DRAWING

An AutoCAD drawing is a file that describes a graphic image. AutoCAD interprets the objects described in the file and draws them on the screen exactly as can be drawn manually.

## COORDINATES

A cartesion coordinate system is used for locating points in the drawing, to position entities, for instance. An X coordinate specifies horizontal location and a Y coordinate specifies vertical location. Thus any point on drawing can be indicated by an X and Y coordinate pair of the form (X,Y). The (0,0) point is normally at the lower left coorner of the drawing.

## DRAWING UNITS

As noted,entities in the drawing are positioned on coordinate points. For example we can draw a line by specifying the coordinates of its two end points. The distance between two points is measured in units. Thus, a line drawn between the points (1,1) and (1,2) is one unit in length.

A unit can correspond to whatever form of measurement our drawing requires. It can be inches, feet, centimeters,angstroms, whatever. Thus, we can draw using "real world" units and eliminate the possibility of scling errors. When the drawing is completed, we can plot it at whatever scale we like. In fact we can plot a given drawing at different scales, eliminating the need for seperate drawings at different scales.

## DISPLAY

The term display has two related meaning usually, displays refers to the portion of the drawing currently being shown. Occasionaly, display means the graphics screen upon which the drawing is shown.

## ZOOMING AND PANNING

The display can be zoomed in or out to magnify or shrink the visible image of the drawing. Zooming in can "blow up" a small portion of the drawing and show more of its details. we can zoom in to draw intricate parts of the drawing with exacting detail and then "back off" to look at the finished drawing AutoCAD's "Zoom ratio" is about ten trillion to one, more than adequate for most applications.

## WINDOW

The graphics screen is used as a window through which we can look at all or part of we drawing, keeping in mind that coordinates refer to fixed locations in the drawing, not to the physical location on the display screen. Therefore the absolute size of a unit remains constant, the points (1,1) and (1,2) are always one unit part, althouh the apparent distance between points on the screen varies with different zoom levels. When you zoom out, the distance between coordinates appears to be small. A line drawn between (1,1) and (1,2) may be only a quqrter inch long measured on the screen. When you zoom in, the distance between coordinates appears larger, so the one-unit line may appear to be several inches long. In both cases, the absolute distance between the coordinates in constant only the screen display changes.

Panning allows us to view a different portion of the drawing without changing its magnification, a line drawn from (1,1) to (1,2) can thus appear in different locations on the screen.

## DRAWING LIMITS AND EXTENTS

The drawing limits are the borders of a rectangle the drawing area, expressed in drawing coordinates.The user car select whatever limits make sense for his drawing. If we want to draw a Printed Circuit Board (PCB) of size 8" x 10" , it is possible to have this size of rectangle.The user can also change these limits as per his requirement.

## THE DRAWING EXTENTS

The drawing extents specify the actual size at the present time. Imagine a rectangle surrounding all the objects in the drawing, the smallest such rectangle defines the drawing extents.

## DISPLAY EXTENTS

AutoCAD keeps track of the current screen location by maintaining a set of boards called display expressed in drawing coordinates. For example, if it is required to display a mangnified view of the center of an 8" x 10" PCB, the display extents might be

Lower Left Corner : (4,3)

Upper Left Corner : (6,5)

Zooming and Panning change the display extents. When these occur, the drawing is generated or redrawn to show only the portion bounded by the new display extents.

## RESOLUTION

Physical resolution refers to the amount of detail that can be represented. The resolution of the display device is specified as "dots x by dots y". Higher resolution means a smoother looking display. This affects the work done on that device, not AutoCAD's internal resolution.

While extending the coordinates, it is possible to snap (lock) them to the nearest point on a grid, which need not be visible. It is completely independent of the resolution of input or output at any time, or turn it off entirely for "free style" drawing.

# AutoLISP

## INTRODUCTION

AutoLISP is a special language used to write instructions carried out by AutoCAD. AutoCAD instruction that have been written in AutoLISP are called LISP routines. LISP routines are contained in ASCII files that are called LISP files. LISP files can have any name, that is a valid DOS file name, and they always have file extensions LISP.

AutoLISP is an implementation of LISP programming language which is a powerful tool for optimising AutoCAD's performances. LISP routines enable the AutoCAD user to automat AutoCAD. LISP routines quickly performs calculations and analysis of data used to generate drawing entities or create new entities by invoking AutoCAD commands directly. The entities created by LISP routines are the same as any created by the user except that with AutoLISP, the process is easier and faster. LISP routines greately enhance the production of complex drawings as well as eliminating unnecessary menu selections and repetitive keyboard entries.

## AutoLISP - FEATURES

AutoLIsp is an implementation of the LISP programming language embeded within the AutoCAD ADE 3 package. AutoLISP users and AutoCAD developers to write Macro programs and functions in a very high level language that is well suited to graphics applications. LISP is easy to learn and use and is flexible.

* LISP excels at working with collections of heterogenous objects in various objects in various sized groups which is precisely the type of information a CAD system like AutoCAD manipulate.

* A LISP interpreter is ideally well suited to understand interaction that characterises the design process.

* LISP is among the easiest language to learn and to master.

* It is the language for research in AI and expert system.

* Also LISP interpreter is fairly easy to implement.

* AutoLISP is enabled using AutoCAD configuration.

**REGULAR AutoLISP**

The requirements of Regular AutoLISP are,

* a computer supported by AutoCAD with atleast 640 KB of memory and hard disk PC-DOS / MS-DOS.

* a matching version of AutoCAD with ADE-3.

**EXTENDED AutoLISP**

The requirements of this Extended AutoLISP are,

* a computer base on the Intel (80286/80386) microprocessor, supported by AutoCAD with at least 640 KB of memory and a hard disk.
* at least 512 KB of IBM AT-style extended memory not for any other purpose.
* PC-DOS / MS-DOS version.
* a matching version of ACAD with ADE-3.

The Extended AutoLISP does work on 80286/80386 models in their 'protected' mode with it residing in the extended memory. Ext-ALISP makes available additional DOS memory to ACAD for I/O paging purposes.

Ext-ALISP does not work on PC/XT class computers since the 8088/8086 CPU's do not allow protected mode. Otherwise there is no difference between Extended ALISP and Regular ALISP.

## ABOUT EXTENDED ALISP

It is implemented as a seperate program Ext LISP that must be run before starting AutoCAD. Extended LISP is a terminated and stay resident (TSR) program that resides partially in Extended memory and communicates with AutoCAD by means of the "acadlx.ovl" overlay file necessary Extended LISP command can be placed in the autoexec.bat file.

Remlisp command at the dos prompt removes EXT.ALISP though making the dos memory occupied by the EXTLISP available memory is as before loading ALISP.

When AutoLISP starts up, it requires two large areas of memory for itself. The first called HEAP is the area in which all functions and variables (also called nodes) are stored.

## DATA TYPES IN AutoLISP

AutoLISP supports several data types such as

- lists
- Symbols - symbol names cannot start with a digit and is not case sensitive to AutoLISP.
- Strings -strings can be of any length. AutoLISP allows dynamic memory allocation for them.
- Real numbers - stored in double precision floating point format providing at least 14 digits (significant) of precision.
- Integers - AutoLISP allow 32 bit signed number but when transfered to AutoCAD only 16 bit transfer occurs.
- File Descriptors.
- AutoCAD entity "names".
- AutoCAD selection sets.
- Subrs (built in function).

## SPECIAL DATA TYPES

- Entity name
- Selection set
- AutoLISP - Evaluation of data
- The evaluator takes a line of user input, evaluates it and returns some result. The following is the process of evaluation.
- Integers, reals, substrings, file pointers and startings evaluate to themselves.

- Symbols evaluate to the value of the current binding.
- Lists are evaluated according to the first element of the list, if it evaluates to
- A list, the list is assumed to be a function definition and the function is evaluated using the values of the remaining elements as arguments
- The name of an internal function ( ) the remaining list elements are passed to the su[illegible] as [illegible] arguments and evaluated by the subr.

When an AutoLISP expression is entered at the AutoCAD command prompt then AutoLISP [illegible] the expression and prints its result and the AutoCAD command reappears.

## LEXICAL NOTATIONS AND CONVENTION

AutoLISP input can be taken in several forms like

- read from ASCII files
- read from a string variable
- read from keyboard from within AutoCAD; but conventions have to remain.
- symbol name can consist of any sequence of printable characters other than delimiting characters (), "" , ; , etc.. These delimiting characters terminate a symbol or numeric constant.

* Expression can be extended to more than one line. .pa
* Literal strings are sequences of characters surrounded by " "; within quotes the control characters can be included.

  e - escape

  n - newline

  t - tab

  r - return

  nnn - octal code

## ERROR HANDLING

If AutoLISP encounters an error during evaluation it brings a message of the form

Error - text where the description of error if ERROR fn is defined (non-nel).AutoLISP executes the function instead of printing the message if the error function is not defined.AutoLISP evaluation stops and trace back display is given.

And variables used,the more heap space will be used.The second area called STACK holds function arguments and partial results.AutoLISP cannot expand its HEAP and STACK space while running under AutoCAD. If enough functions and will display the error message.AutoLISP will not function until more memory is made available and AutoCAD is executed again.

Default sizes for the HEAP and STACK areas are:

HEAP - 5000 bytes.

STACK - 5000 bytes The amount of memory can be altered and the sum of the two areas cannot exceed 45000 bytes.

## AutoLISP PROGRAMMING

AutoLISP program is written for the three types of modification. The procedure for developing the program is given in the figure. First the computer is initialised into the AutoCAD subdirectory. By means of "SHELL" command , we get into the system where the program is being input.

The basic parameters of the component is given. The other dimensions are defined as the function of "BASIC" dimensions. The AutoCAD commands are written in the AutoLISP format using those commands the drafting sequence is incorporated in the program.

If the program is lengthy, same part of the input is stored in the virtual memory.

The program is then loaded. The input parameters are obtained from tabulation using "TABLE SEARCH" command. Then the department parameters are evaluated. The drafting proceeds module by module and the output is obtained.

```
(defun c:main ()
(textscr)
(princ "\n  \n \n \n \n \n \n \n \n \n \n ")
(princ "\n\t\t *** MAIN MENU ***")
(princ \n \n)
(princ "\n\n\t1. Exit")
(princ \n \n)
(princ "\n\n\t2. Design")
(princ \n \n)
(princ "\n\n\t3. Exisiting Drawings")
(setq
          opt (getint "\n\n\n\n\t Enter selection : ")
)


(while
              (or (> opt 3)(< opt 1) (= opt nil)
)

(princ "\n Invalid selection \n\t Please Try again \007")
(textscr)
(princ "\n \n \n \n \n \n \n \n ")
(princ "\n\t\t *** MAIN MENU ***")
(princ \n \n)
(princ "\n\n\t1. Exit")
(princ \n \n)
(princ "\n\n\t2. Design")
(princ \n \n)
(princ "\n\n\t3. Exisiting Drawings")
(setq
          opt (getint "\n\n\n\n\t Enter selection : "))
)

(cond
              ((= opt 1) (command "script" "exit"))
              ((= opt 2) (command "menu" "sur"))
              ((= opt 3) (seldwg))
)
)

(defun seldwg()
((textscr)
(princ "\n  \n \n \n \n \n \n \n \n \n \n ")
(princ "\n\t\t *** SUB MENU  (DRAWINGS) ***")
(princ \n \n)
(princ \n \n \t1.DIESET)
(princ \n \n \t2.STRIPPER)
(princ \n \n \t3.GUIDE BLOCK)
(princ \n \n \t4.HARDENED PLATE (FOR STAMP HOLDER))
(princ \n \n \t5.HARDENED PLATE)
(princ \n \n \t6.SETTING BLOCK I)
(princ \n \n \t7.SETTING BLOCK II)
(princ \n \n \t8.STAMP HOLDER)
```

```
(setq opt1
            (getint "\n\n\n\n\tENTER SELECTION)
)
(while
            (or (> opt1 9)(< opt 1) (= opt nil)
)
(princ "\n Invalid selection \n\t Please Try again \007")
(textscr)
(princ "\n  \n \n \n \n \n \n \n \n \n \n ")
(princ "\n\t\t *** SUB MENU  (DRAWINGS) ***")
(princ \n \n)
(princ \n \n \t1.DIESET)
(princ \n \n \t2.STRIPPER)
(princ \n \n \t3.GUIDE BLOCK)
(princ \n \n \t4.HARDENED PLATE (FOR STAMP HOLDER))
(princ \n \n \t5.HARDENED PLATE)
(princ \n \n \t6.SETTING BLOCK I)
(princ \n \n \t7.SETTING BLOCK II)
(princ \n \n \t8.STAMP HOLDER)
(princ \n \n \t9.CLAMPING PLATE)
(setq opt1 (princ "\n\n\n\n\t ENTER SELECTION: ")
(setq ppp1 (getpoint "\nPick a point ")
(cont
    ((= opt 1) (command "insert" "dieset" ppp1 "" "" "" )
    ((= opt 2) (command "insert" "strip" ppp1 "" "" "" )
    ((= opt 3) (command "insert" "gblock" ppp1 "" "" "" )
    ((= opt 4) (command "insert" "HplateI" ppp1 "" "" "" )
    ((= opt 5) (command "insert" "HplateII" ppp1 "" "" "" )
    ((= opt 6) (command "insert" "seblockI" ppp1 "" "" "" )
    ((= opt 7) (command "insert" "seblockII" ppp1 "" "" "" )
    ((= opt 8) (command "insert" "spholder" ppp1 "" "" "" )
    ((= opt 9) (command "insert" "cplate" ppp1 "" "" "" )
)
)
```

```
@ echo off
cls
acad washer start
cls
```

```
2
washer
(load "main")
main
```

```
quit y Ø
```

```
***buttons
;
$p1=*


***pop1
[Tools]
[Washer ...      ]^C^C$i=wash1 $i=*


***screen
[AutoCAD]^c^c(load "men")
[]
[Washer]^C^C$i=wash1 $i=*

***icon
 **wash1
 [Select  WASHER type Please...]
 [nw1]^C^C(load  "tha");VSLIDE;NW1;(PROMPT "  Press Any  Key
 To  Continue....");(grread);redraw;nw1;(load "diedesi");
 (load "diedra");
 [nw2]^C^C(load  "tha");VSLIDE;NW2;(PROMPT "  Press Any  Key
 To Continue....");(grread);redraw;nw2;(load "diedesi");
 (load "diedra");
 [nw3]^C^C(load "tha");VSLIDE;NW3;(PROMPT "  Press Any  Key
 to Continue....");(grread);redraw;nw3;(load "diedesi");
 (load "diedra");
 [ NEXT ]^c^c$i=wash2 $i=*


 **wash2
 [Select  WASHER type Please...]
 [nw4]^C^C(load  "tha");VSLIDE;NW4;(PROMPT "  Press Any  Key
 To Continue....");(grread);redraw;nw4;(load "diedesi");
 (load "diedra");
 [nw5]^C^C(load  "tha");VSLIDE;NW5;(PROMPT "  Press Any  Key
 to Continue....");(grread);redraw;nw5;(load "diedesi");
 (load "diedra");
 [nw6]^C^C(load  "tha");VSLIDE;NW6;(PROMPT "  Press Any  Key
 To Continue....");(grread);redraw;nw6;(load "diedesi");
 (load "diedra");
 [ NEXT ]^c^c$i=wash3 $i=*


 **wash3
 [Select  WASHER type Please...]
 [nw7]^C^C(load  "tha");VSLIDE;NW7;(PROMPT "  Press Any  Key
 To Continue....");(grread);redraw;nw7;(load "diedesi");
 (load "diedra");
 [nw8]^C^C(load  "tha");VSLIDE;NW8;(PROMPT "  Press Any  Key
 to Continue....");(grread);redraw;nw8;(load "diedesi");
 (load "diedra");
 [nw9]C^C(load  "tha");VSLIDE;NW9;(PROMPT "  Press Any  Key
 To Continue....");(grread);redraw;nw8;(load "diedesi");
 (load "diedra");
```

```
[ NEXT ]^c^c$i=wash4 $i=*


**wash4
[Select  WASHER type Please...]
[nw10]^C^C(load  "tha");VSLIDE;NW10;(PROMPT "  Press Any  Key
To Continue....");(grread);redraw;nw10;(load "diedesi");
(load "diedra");
[nw11]^C^C(load  "tha");VSLIDE;NW11;(PROMPT "  Press Any  Key
to Continue....");(grread);redraw;nw11;(load "diedesi");
(load "diedra");
[nw12]^C^C(load  "tha");VSLIDE;NW12;(PROMPT "  Press Any  Key
To Continue....");(grread);redraw;nw12;(load "diedesi");
(load "diedra");
[ NEXT ]^c^c$i=wash5 $i=*


**wash5
[Select  WASHER type Please...]
[nw13]^C^C(load  "tha");VSLIDE;NW13;(PROMPT "  Press Any  Key
To Continue....");(grread);redraw;nw13;(load "diedesi");
(load "diedra");
[nw14]^C^C(load  "tha");VSLIDE;NW14;(PROMPT "  Press Any  Key
to Continue....");(grread);redraw;nw14;(load "diedesi");
(load "diedra");
[nw15]^C^C(load  "tha");VSLIDE;NW15;(PROMPT "  Press Any  Key
To Continue....");(grread);redraw;nw15;(load "diedesi");
(load "diedra");
[ NEXT ]^c^c$i=wash6 $i=*


**wash6
[Select  WASHER type Please...]
[nw16]^C^C(load  "tha");VSLIDE;NW16;(PROMPT "  Press Any  Key
To Continue....");(grread);redraw;nw16;(load "diedesi");
(load "diedra");
[quit]^c^cquit; y;
```

```
;******************************************************************
;           COMPUTER  AIDED  DESIGN  AND  FABRICATION  OF   A
;                    PROGRESSIVE DIESET FOR THRUST WASHERS
;
;******************************************************************

    (defun dtr (a) (/ (* pi a) 180))
    (defun rtd (a) (/ (* 180 a) pi))

    (defun c:nw1()
    (load "lowp1")
    (load "upp1")
    )

    (defun c:nw2()
    (load "lowp1")
    (command "arc" 16 "e" 11 cpt1
    )
    )

    (defun c:nw3()
    (load "lowp3")
    (load "upp3")
    )

    (defun c:nw4()
    (load "lowp3")
    (load "upp2")
    )

    (defun c:nw5()
    (load "lowp1")
    (load "upp2")
    )

    (defun c:nw6()
    (load "lowp1")
    (load "upp4")
    )

    (defun c:nw7()
    (load "lowp1")
    (load "upp3")
    )

    (defun c:nw9()
    (load "lowp3")
    (load "upp4")
    )
```

```
(defun c:nw10()
(load "lowp3")
(load "upp1")
)

(defun c:nw11()
(load "lowp3")
(command "arc" 16 "e" 11 cpt1)
)

(defun c:nw12()
(load"lowp4")
(load "upp1")
)

(defun c:nw13()
(load"lowp4")
(load "upp4")
)

(defun c:nw14()
(load"lowp4")
(load "upp2")
)

(defun c:nw15()
(load"lowp4")
(load "upp3")
)

(defun c:nw16()
(load"lowp4")
(command "arc" 16 "e" 11 cpt1)
)
```

```
(setq R (getDIST "\n Enter the outer radius :")
     r1 (getDIST "\n Enter the inner radius :")
     r2 (getDIST "\n Enter the offset radius :")
      z (getreal "\n Enter the value of z :")
   cpt1 (list 90.00 100.00)
     l1 (list (- 90.00 R) 100.00)
     l6 (list (+ 90.00 R) 100.00)
     p2 (list (- 90.00 (- R z)) 100.00)
     p1 (list (+ 90.00 (- R z)) 100.00)
     l1 (list (car cpt1) (+ (cadr cpt1) r1))
 )

(command "arc"  p1 "e" p2 "r" r2)
(setq elist (entget (entlast)))
(entdel (entlast))
(setq cpt2 (cdr (assoc 10 elist)))
  (setq x (+ r1 z)
        y (sqrt (ABS (- (EXPT R 2) (EXPT x 2))))
       p5 (list (+ (car cpt2) x) (+ (cadr cpt2) y))
       p4 (list (- (car cpt2) x) (+ (cadr cpt2) y))
   )
(command "arc" p4 "e" p2 "r" r2
         "arc" p1 "e" p5 "r" r2
         "arc" p5 "e" p4 cpt1
         "line"  l1 p2 ""
         "line"  p1 l6 ""
)
```

```
  (setq r (getDIST "\n ENTER THE OUTER RADIUS : ")
       r1 (getDIST "\n ENTER THE INNER RADIUS: ")
       zz (getDIST "\n ENTER THE END ANGLE: ")
     cpt1 (list 90.00 100.00)
        x (* (/ (sin  zz) (cos  zz)) (/ (- r r1) 2))
       l1 (list (- 90.00 r) (- 100.00 x))
       l2 (list (- 90.00 r1) (+ 100.00 x))
       l3 (list (+ 90.00 r1) (- 100.00 x))
       l6 (list (+ 90.00 r) (+ 100.00 x))
       l1 (list (car cpt1) (+ (cadr cpt1) r1))
  )

  (command "arc" l3 "e" l2 cpt1
           "line" l6 l3 ""
           "line" l2 l1 ""
  )
```

```
(setq
     lw (getDIST "\n ENTER THE LEG WIDTH : ")
     lh (getDIST "\n ENTER THE LEG HEIGHT :")
     h  (getDIST "\n ENTER HT. OF WASH (RT) : ")
     h1 (getDIST "\n ENTER HT. OF WASH (LT) : ")
)

(setq cpt1(list 90.00 100.00)
      j (sqrt (- (expt r 2) (expt h 2)))
      i (sqrt (- (expt r 2) (expt h1 2)))
     ua (list (- 90.00 i) (+ 100.00 h1))
     ub (list (- 90.00 lw) (+ 100.00 h1))
     uc (list (car ub) (+ (+ 100.00 h) lh))
     ud (list 90.00 (cadr uc))
     ue (list 90.00 (+ 100.00 h))
     uf (list (+ 90.00 j) (cadr ue))
      l1 (list (- 90.00 r) 100.00)
      l6 (list (+ 90.00 r) 100.00)
 )

 (command         "line" uf ue ""
                  "line" ue ud ""
                  "line" ud uc ""
                  "line" uc ub ""
                  "line" ub ua ""
                  "arc" l6 "e" uf cpt1
                  "arc" ua "e" l1 cpt1
 )



  (setq
           h (getDIST "\n ENTER THE HT. OF WASH: ")
           z (sqrt (- (expt r 2) (expt h 2) ))
          ua (list (- 90.00 z) (+ 100.00 h))
          ub (list (+ 90.00 z) (cadr ua))
  )

 (command "arc" l6 "e" ub cpt1
          "arc" ua "e" l1 cpt1
          "line" ub ua ""
 )
```

```
(setq r (getDIST "\n ENTER THE OUTER RADIUS : ")
     r1 (getDIST "\n ENTER THE INNER RADIUS: ")
     zz (getDIST "\n ENTER THE END ANGLE (give zero):
   cpt1 (list 90.00 100.00)
      x (* (/ (sin  zz) (cos  zz)) (/ (- r r1) 2))
     l1 (list (- 90.00 r) (- 100.00 x))
     l2 (list (- 90.00 r1) (+ 100.00 x))
     l3 (list (+ 90.00 r1) (- 100.00 x))
     l6 (list (+ 90.00 r) (+ 100.00 x))
     l1 (list (car cpt1) (+ (cadr cpt1) r1))
)

(command "arc" l3 "e" l2 cpt1
         "line" l6 l3 ""
         "line" l2 l1 ""
)
```

```
(setq r (getDIST "\n ENTER THE OUTER RADIUS : ")
     r1 (getDIST "\n ENTER THE INNER RADIUS: ")
     zz (getDIST "\n ENTER THE END ANGLE (give zero):
   cpt1 (list 90.00 100.00)
      x (* (/ (sin  zz) (cos  zz)) (/ (- r r1) 2))
     l1 (list (- 90.00 r) (- 100.00 x))
     l2 (list (- 90.00 r1) (+ 100.00 x))
     l3 (list (+ 90.00 r1) (- 100.00 x))
     l6 (list (+ 90.00 r) (+ 100.00 x))
     l1 (list (car cpt1) (+ (cadr cpt1) r1))
)

(command "arc" l3 "e" l2 cpt1
         "line" l6 l3 ""
         "line" l2 l1 ""
)
```

```
(setq
       lw (getDIST "\n ENTER THE LEG WIDTH : ")
       lh (getDIST "\n ENTER THE LEG HEIGHT: ")
        h (getDIST "\n ENTER THE HT. OF WASH: ")
        z (sqrt (- (expt r 2) (expt h 2)))
       ua (list (- 90.00 z) (+ 100.00 h))
       ub (list (+ (- 90.00 z) (/ lw 2)) (cadr ua))
       uc (list (car ub) (+ (cadr ua) lh))
       ud (list (- (+ 90.00 z) (/ lw 2)) (cadr uc))
       ue (list (car ud) (cadr ua))
       uf (list (+ 90.00 z) (cadr ua))
)

(command
            "arc" 16 "e" uf cpt1
            "line" ua ub ""
            "line" ub uc ""
            "line" uc ud ""
            "line" ud ue ""
            "line" ue uf ""
            "arc" ua "e" 11 cpt1
 )
```

```
(defun c:die ()

 (setq t (getdist "\nThickness of Washer : "))
         (if (> t 4.25)
         (progn
         (prompt " ***INVALID DATA ***")
         (prompt "thickness should be less than or equal to
                                              4.25")

         (setq t(getdist "\n Thickness of washer: "))
         )
   )

   (setq ga  (getdist "\nGrinding Allowance : ")
        peri (+ (* pi 2 r ) (* 2 pi r1) (* 2 (- r r1)))
   )
   (if (and (<= t 2.5)(<= peri 5Ø)) (diethick))
       (cond
       ((<= peri 75) (setq diet (* 7 1.25)))
       ((<= peri 15Ø) (setq diet (* 7 1.5)))
       ((<= peri 3ØØ) (setq diet (* 7 1.75)))
        ((> peri 3ØØ) (setq diet (* 7 2)))
   )

   (setq    ma (* 2 diet)
         diet1 (+ diet ga)
             d (* 2 r)
           wid (+ (* 2 ma) d)
           len (+ (* 2 ma) r)
           wa  (* wid len)
           )
;************************Grouping/Working area***********

   (cond
            ((<= wa 45ØØ)  (setq ss  1Ø ns 2))
            ((<= wa 63ØØ) (setq ss  1Ø ns 3))
            ((> wa 63ØØ) (setq ss  12 ns 4))
     )


     (pitch)
     (diethick)
     (worstd)
     (cegra)
     (tonn)
     (cle)
     (strip)
     (printing)
     )
   ;***************Select data *******

    (defun diethick ()
    (setq chk "Working"
        fpt (open "sur.dat" "r")
     )
     (while (= chk  "Working")
             (setq rea1 (read-line fpt)
                  dat1 (atof  rea1)
                  rea2 (read-line fpt)
              )
     (cond
            ((= dat1 t)
            (setq diet (atof rea2)
                chk "Over")(close fpt))
                ((=   dat1  nil)(princ   "\nCannot  Find   Data
                                       File..")(close fpt))

                (T (prompt "\nReading"))
              )
         )
       )

       ;************** WorSurstd********

       (defun worstd ()
          (setq chk "Working"
                lno -1
               fpt1 (open "four.dat" "r")
          )

        (while (= chk "Working")
        (setq rea1 (atof (read-line fpt1))
             rea2 (atof (read-line fpt1))
              lno (1+ lno)
         )
         ;(princ rea1)(terpri)(princ rea2)(terpri)(getdist)
          (if (or (= rea1 nil) (= rea2 nil) )
           (progn
                  (princ "\nWorking surface Overflow ")
                  (close fpt1)
                  (setq chk "Over-e")
           )
          )
;   **************** if starts***************

           (if (and (<= wid rea1 ) (<= len rea2))
               (progn
                    (setq
                         a1 (atof (read-line fpt1) )
                         a2 (atof (read-line fpt1) )
                  b2 (atof (read-line fpt1) )
                  e  (atof (read-line fpt1) )
                  d1 (atof (read-line fpt1) )
                  d2 (atof (read-line fpt1) )
                  e1 (atof (read-line fpt1) )
                  e3 (atof (read-line fpt1) )
                  c1 (atof (read-line fpt1) )
                  c2 (atof (read-line fpt1) )
                  h  (atof (read-line fpt1) )
                  l  (atof (read-line fpt1) )
            chk "Over"
           )

                   (close fpt1)

          )

;*********if end-*************

      (progn

                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)
                (read-line fpt1)

           )

          )

        )

       )



   ;******
    (defun cegra()
              (setq y (/ (* 4  (+ (expt r 2) (expt r1 2) (*
                              r r1))) (* 3 pi (+ r r1)))

                   x (car cpt1)
               )
             (setq cog (list x y)
             )
      )
    (defun tonn()

            (setq
                   ton (* 38.5 peri t)
              )
      )

;****************
    (defun pitch()

      (if (< t Ø.8)
          (setq b Ø.8)
       )

      (if  (and (> t Ø.8) (< t 3.2))
           (setq b t)

       )

       (if (> t 3.2)
           (setq b 3.2)
       )

              (setq pit(+ r1 3.2)
              )
       )

  ;*****************
        (defun cle()
        (setq le (* Ø.Ø7 t))
        )

  ;****************
        (defun strip()

         (setq clear 12.7
               w (+ r (* 2 clear))
             spt (/ (+ (/ w 3) (* 16 t)) 8)
             hts (* 1.5 t)
          )
         )


   ;****************
         (defun printing()
                                                            :")(princ
         (prompt "Thickness of washer                      t)(terpri)

                                                            :")(princ
         (prompt "Die plate thickness                    diet)(terpri)

                                                            :")(princ
         (prompt "Perimeter                              peri)(terpri)
(prompt "Margin on either side                      :")(princ
                                                   ma)(terpri)

(prompt "width of the die                           :")(princ
                                                  wid)(terpri)

(prompt "Length of the die                          :")(princ
                                                  len)(terpri)

(prompt "working area                               :")(princ
                                                   wa)(terpri)

(prompt "Centre of gravity of the washer            :")(princ
                                                    t)(terpri)

(prompt "Press Tonnage                              :")(princ
                                                 ton) (terpri)

(prompt "Pitch distance                             :")(princ
                                                  pit)(terpri)

(prompt "clearance between punch and die            :")(princ
                                                   le)(terpri)

(princ (strcat "\nScrew size for fixing the die =    M"
                                                     (itoa ss)
                     "\nNo. of screws =   "(itoa ns)
  )
  )
  (prompt   "stripper   plate   thickness:   ")   (princ
                                                 spt)(terpri)

  (prompt  "Height  of stock strip channel :  ")  (princ
                                                 hts)(terpri)
  (setq ;lno (/ (+ 13 lno) 15)
        fpt2 (open "four1.dat" "r")
  )

  (princ "\n")

  (repeat 4
  (princ (read-line fpt2))
  (princ "\n")
  )

  (repeat (fix lno)
  (read-line fpt2)
  )
  (princ (read-line fpt2))
  (close fpt2)

  )
```

```
(defun dra()
(setq cg (list (car cpt1) (+ (cadr cpt1)(/ r 3)))
      q1 (list  (- (car cg) 107.85)(- (cadr cg) 95.25))
      q2 (list (car q1) (+ (cadr q1) 177.8))
      q3 (list  (+ (car q2) 12.7) (cadr q2))
      q4 (list (car q3)  (cadr q1))
      q5 (list  (+ (car q4) 190.5) (cadr q1))
      q6 (list (car q5)  (cadr q3))
      q7 (list  (+ (car q6) 12.7) (cadr q6))
      q8 (list  (car q7) (cadr q5))
      q9 (list (+ (car cg) 79.375)  (+ (cadr q4) 12.7))
     q10 (list  (car q9) (- (cadr q3) 12.7))
     q11 (list (- (car cg) 79.375)  (cadr q9))
     q12 (list  (car q11) (cadr q10))
     q13 (list (+ (car q8) 254) (cadr q8))
     q14 (list (car q13) (cadr q7))
     q15 (list (+ (car q14) 20.6375) (cadr q7))
     q16 (list (+ (car q14) 50.8) (cadr q7))
     q17 (list (car q16) (cadr q8))
     q18 (list (car q15) (cadr q8))
     q19 (list (car q16) (+ (cadr q9) 4.7625))
     q20 (list (- (car q19) 19.05) (cadr q19))
     q21 (list (car q20) (- (cadr q9) 4.7625))
     q22 (list (car q16) (cadr q21))
     q23 (list (car q16) (cadr q10))
     q24 (list (car q14) (- (cadr q14) 381))
     q25 (list (+ (car q24) 20.6375) (cadr q24))
     q26 (list (car q25) (- (cadr q25) 12.7))
     q27 (list (car q16) (cadr q26))
     ax1 (list (car q24) (- (cadr q24) 107.95))
     ax2 (list (car q27) (cadr ax1))
     q28 (list (car q16) (- (cadr q27) 11.1125))
     q29 (list (car q24) (+ (cadr ax1) r))
     q30 (list (car q24) (- (cadr ax1) r))
     q31 (list (car q27) (+ (cadr ax1) r))
     q32 (list (car q27) (- (cadr ax1) r))
     q33 (list (car q27) (- (cadr ax2) 200))
  )

  (command
            "pline" q1 q2 q7 q8 q1 ""
            "line" q5 q6 ""
            "line" q3 q4 ""
            "circle" q9 "d" 9.525
            "copy" "l" "" "M" q9 q10 q11 q12 ""
            "circle" q9 "d" 8.75
            "copy" "l" "" "M" q9 q10 q11 q12 ""
            "point" q9
            "point" q10
            "point" q11
            "point" q12
            "point" cg
```

```
                "pline" q14 q16 q17 q13 q14 ""
                "line" q15 q18 ""
                "pline" q19 q20 q21 q22 ""
                "copy" q21 "" q18 q23
                "pline" ax1 q24 q25 q26 q27 ax2 ""
                "copy" q21 "" q19  q28
                "mirror" q24 q28 "" ax1 ax2 ""
                "linetype" "set" center ""
                "ltscale" 100
                "line" q29 q31 "´
                "line" q30 q32 ""
            )
       )
```

```
(defun alw1()
    (setq
           s1 (list (car q14) (cadr 16))
           s3 (list (car q14) (cadr 11))
           s4 (list (car q16) (cadr 16))
           s6 (list (car q16) (cadr 11))
           p1 (list (+ (car q16) 50.8) (cadr 16))
           p3 (list (car p1) (cadr 11))
      )
  )


  (defun aup1()
     (setq
          s10 (list (car q14) (cadr uc))
          s12 (list (car q14) (cadr ub))
          s13 (list (car q16) (cadr uc))
          p11 (list (+ (car q16) 50.8) (cadr ub))
          s11 (list (car q16) (cadr uc))
          p10 (list (+ (car q16) 50.8) (cadr uc))

     )
   )


    (defun dielw1()
    (alw1)
    (command
               "linetype" "set" center ""
               "ltscale" 100
               "line" s1 s4 ""
               "line" s3 s6 ""
               "linetype" "set" continuous ""

     )
    )


    (defun dieup1()
    (aup1)
      (command
                "linetype" "set" center ""
                "ltscale" 100
                "line" s10 s11 ""
                "linetype" "set" continuous ""

      )
    )
```

```
(defun punlw1()
(alw1)
(command        "line" s4 p1 ""
                "line" s6 p3 ""

 )
 )


 (defun punup1()
 (aup1)
 (command        "line" s11 p10 ""

 )
  )


  (defun aup2()
  (setq
         s10 (list (car q14) (cadr ua))
         s11 (list (car q16) (cadr ua))
         s12 (list (car q14) (cadr ub))
         s13 (list (car q16) (cadr ub))
         s14 (list (car q14) (cadr uc))
         s15 (list (car q16) (cadr uc))
         s16 (list (car q14) (cadr ud))
         s17 (list (car q16) (cadr ud))
         p10 (list (+ (car q16) 50.8) (cadr ua))
         p11 (list (+ (car q16) 50.8) (cadr ub))
         p12 (list (+ (car q16) 50.8) (cadr uc))
         p13 (list (+ (car q16) 50.8) (cadr ud))
    )
    )


     (defun alw2()
     (setq
            s1 (list (car q14) (cadr l1))
            s2 (list (car q16) (cadr l1))
            s3 (list (car q14) (cadr l6))
            s4 (list (car q16) (cadr l6))
            p1 (list (+ (car q16) 50.8) (cadr l1))
            p2 (list (+ (car q16) 50.8) (cadr l6))
        )
      )


      (defun dielw2()
      (alw2)
      (command        "linetype" "set" center ""
          "line" s1 s2 ""
          "line" s3 s4 ""
)
)


(defun dieup2()
(aup2)
(command
          "linetype" "set" center ""
          "ltscale" 1ØØ
          "line" s1Ø s11 ""
          "line" s12 s13 ""
          "line" s14 s15 ""
          "line" s16 s17 ""
          "linetype" "set" continuous ""
 )
 )


 (defun punlw2()
 (alw2)
 (command
           "line" s2 p1 ""
           "line" s4 p2 ""
  )
  )

  (defun punup2()
  (aup2)
  (command
            "line" s11 p1Ø ""
            "line" s13 p11 ""
            "line" s15 p12 ""
            "line" s17 p13 ""
   )
   )



   (defun alw3()

   (setq
            s1 (list (car q14) (cadr l1))
            s2 (list (car q16) (cadr l1))
            s3 (list (car q14) (cadr l6))
            s4 (list (car q16) (cadr l6))
            p1 (list (+ (car q16) 5Ø.8) (cadr l1))
            p2 (list (+ (car q16) 5Ø.8) (cadr l6))
      )
      )
```

```
(defun dielw3()
(alw3)
(command
        "linetype" "set" center ""
        "ltscale" 100
        "line" s1 s2 ""
        "line" s3 s4 ""
        "linetype" "set" continuous ""
)
)


 (defun punlw3()
 (alw3)
 (command
          "line" s2 p1 ""
          "line" s4 p2 ""
  )
 )


 (defun aup3()
  (setq
          s10 (list (car q14) (cadr ua))
          s11 (list (car q16) (cadr ua))
          s12 (list (car q14) (cadr uc))
          s13 (list (car q16) (cadr uc))
          p10 (list (+ (car q16) 50.8) (cadr ua))
          p11 (list (car p10) (cadr uc))
  )
 )


  (defun dieup3()
  (aup3)
  (command
                  " linetype" "set" center ""
                  "ltscale" 100
                  "line" s10 s16 ""
                  "line" s12 s13 ""
                  "linetype" "set" continuous ""


    )
   )


   (defun punup3()
   (aup3)
   (command
                     "line" s11 p10 ""
                     "line" s13 p11 ""

    )
     )
```

```
(defun aup4()
(setq
           s10 (list (car q14) (cadr ua))
           s11 (list (car q16) (cadr ua))
           s12 (list (car q14) (cadr ub))
           s13 (list (car q16) (cadr ub))
           p10 (list (+ (car q16) 50.8) (cadr ua))
           p11 (list (+ (car q16) 50.8) (cadr ub))
  )
)


(defun dieup4()
(command
           "linetype" "set" center ""
           "ltscale" 100
           "line" s10 s11 ""
           "line" s12 s13 ""
           "linetype" "set" continuous ""
  )
)


(defun punup4( )
(command
           "line" s11 p10 ""
           "line" s13 p11 ""
 )
)


(defun addpnw1()
(setq
         name (getstring "\n Block name : "))
(punup1)
(punlw1)
(command
           "line" p1 p10 ""
           "rotate" c q1 q33 90
  )
   )


(defun adddnw1()
(setq
           name (getstring "\n Block name : "))
(dieup1)
(dielw1)
(command
           "rotate" c q1 q33 90 )
 )
```

```
(defun adddnw2()
(setq
        name (getstring "\n Block name : "))
(dielw2)
(setq
        rr1(list (car q14) (cadr ll))
        rr2 (list (car q16) (cadr LL))
)
(command
            "linetype" "set" center ""
            "ltscale" 100
            "line" rr1 rr2 ""
            "linetype" "set" continuous "")
(command
            "rotate" c q1 q33 90 )
)


(defun addpnw2()
(setq
          name (getstring "\n Block name : "))
(punlw2)
(setq
           rr10(list (+ (car q16) 50.8) (cadr ll)))
(command
           "line" p1 p2 "")
(command
           "rotate" c q1 q33 90 )
)


(defun adddnw3()
(setq
            name (getstring "\n Block name : "))
(dielw3)
(dieup3)
(command
            "rotate" c q1 q33 90 )
)


(defun addpnw3()
(setq
            name (getstring "\n Block name : "))
(punlw3)
(punup3)
(command
            "line" p1 p10 ""
            "rotate" c q1 q33 90
  )
  )
```

```
(defun adddnw4()
(setq
           name (getstring "\n Block name : "))
(dielw3)
(dieup2)
(command
            "rotate" c q1 q33 9Ø )
 )


(defun addpnw4()
(setq
           name (getstring "\n Block name : "))
(punlw3)
(punup2)
(command
           "line" p1 p1Ø ""
           "rotate" c q1 q33 9Ø
)
)

(defun adddnw5()
(setq
           name (getstring "\n Block name : "))
(dielw1)
(dieup2)
(command
           "rotate" c q1 q33 9Ø )
)

(defun addpnw5()
(setq
           name (getstring "\n Block name : "))
(punlw 1)
(punup2)
(command
           "line" p1 p1Ø ""
           "rotate" c q1 q33 9Ø
 )
)

(defun adddnw6()
(setq
           name (getstring "\n Block name : "))
(dielw1)
(dieup4)
(command
           "rotate" c q1 q33 9Ø )
)

(defun addpnw6()
(setq
(punlw1)
(punup4)
(command
                "line" p1 p10 ""
                "rotate" c q1 q33 90
)
)

(defun adddnw7()
(setq
                name (getstring "\n Block name : "))
(dielw1)
(dieup3)
(command
                "rotate" c q1 q33 90 )
)

(defun addpnw7()
(setq
                name (getstring "\n Block name : "))
(punlw1)
(punup3)
(command
                "line" p1 p10 ""
                "rotate" c q1 q33 90
)
)

(defun adddnw8()
(setq
                name (getstring "\n Block name : "))
(dielw2)
(dieup4)
(command
                "rotate" c q1 q33 90 )
)

(defun addpnw8()
(setq
                name (getstring "\n Block name : "))
(punlw2)
(punup4)
(command
                "line" p1 p10 ""
                "rotate" c q1 q33 90
)
)

(defun adddnw9()
(setq
                name (getstring "\n Block name : "))
(dielw3
(command
                       "rotate" c q1 q33 90 )
)

(defun addpnw9()
(setq
                       name (getstring "\n Block name : "))
(punlw3)
(punup4)
(command
                       "line" p1 p10 ""
                       "rotate" c q1 q33 90
)
)

(defun adddnw10()
(setq
                       name (getstring "\n Block name : "))
(dielw3)
(dieup1)
(command
                       "rotate" c q1 q33 90 )
)

(defun addpnw10()
(setq
                       name (getstring "\n Block name : "))
(punlw3)
(punup1)
(command
                       "line" p1 p10 ""
                       "rotate" c q1 q33 90
)
)

(defun adddnw11()
(setq
                       name (getstring "\n Block name : "))
(dielw3)
(setq
                       rr1 (list (car q14) (cadr ll))
                       rr2 (list (car q16) (cadr LL)))
(command
                       "linetype" "set" center ""
                       "ltscale" 100
                       "line" rr1 rr2 ""
                       "linetype""set" continuous "")
(command
                       "rotate" c q1 q33 90 )
)

(defun addpnw11()
(setq
                       name (getstring "\n Block name : "))
```

```
(punlw3)
(setq
             rr10 (list (+ (car q16) 50.8) (cadr l1)))
(command
                "line" p1 p2 "")
(command
                "rotate" c q1 q33 90 )
)

(defun adddnw12()
(setq
                 name (getstring "\n Block name : "))
(dielw4)
(dieup1)
(command
                 "rotate" c q1 q33 90 )
)

(defun addpnw12()
(setq
                 name (getstring "\n Block name : "))
(punlw4)
(punup1)
(command
                 "line" p1 p10 ""
                 "rotate" c q1 q33 90
)
)

(defun adddnw13()
(setq
                 name (getstring "\n Block name : "))
(dielw4)
(dieup4)
(command
                 "rotate" c q1 q33 90 )
)

(defun addpnw13()
(setq
                 name (getstring "\n Block name : "))
(punlw4)
(punup4)
(command
                 "line" p1 p10 ""
                 "rotate" c q1 q33 90
)
)

(defun adddnw14()
(setq
                 name (getstring "\n Block name : "))
(dielw4)
```

```
(command                "rotate" c q1 q33 9Ø )

)

(defun addpnw14()
(setq                   name (getstring "\n Block name : "))

(punlw4)
(punup2)
(command                "line" p1 p1Ø ""
                        "rotate" c q1 q33 9Ø

)
)

(defun adddnw15()
(setq name              (getstring "\n Block name : "))

(dielw4)
(dieup3)
(command                "rotate" c q1 q33 9Ø )

)

(defun addpnw15()
(setq                   name (getstring "\n Block name : "))

(punlw4)
(punup3)
(command                "line" p1 p1Ø ""
                        "rotate" c q1 q33 9Ø

)
)

(defun adddnw16()
(setq                   name (getstring "\n Block name : "))

(dielw4)
(setq                   rr1 (list (car q14) (cadr ll))
                        rr2 (list (car q16) (cadr LL)))

(command                "linetype" "set" center ""
                        "ltscale" 1ØØ
                        "line" rr1 rr2 ""
                        "linetype""set" continuous "" )

(command                "rotate" c q1 q33 9Ø )

)


(defun addpnw16()
(setq                   name (getstring "\n Block name : "))
```

```
(punlw4)
(setq              rr1Ø (list (+ (car q16) 5Ø.8) (cadr l1)))
(command "line" p1 p2 "")
(command "rotate" c q1 q33 9Ø )
)
```

## ABOUT THE PROGRAM

This software is basically developed with the view to ease the design procedure and can successfully can check the values obtained by design values with the actual drawing of the punch and dieplate.

First, the main menu is loaded. In the main menu two options are left to the user. One is the design and another is the existing drawing. If the user selects the existing drawings he can select the part drawings stored in the file. There are nine part drawings in the existing drawing, they are

a) Stripper plate

b) Guide plate

c) Setting block for die & punch

d) Setting block for butting

e) Hardened plate for placing above dieset

f) Hardened plate for stamp holder

g) Clamping plate

h) Stamp holder

i) Dieset

If the user selects the design in main menu, the submenu is executed and submenu is displayed which has the Washer types for design, Acad menu and Main menu and Quit.

The Acad menu is listed for the user to make use of Autocad commands during design. For instance if the user wasnts to plot the drawing he can make use of the Acad menu.

The Main menu is listed to goback from the submenu to the main menu for having a second option. Quit is listed to quit to Autocad main menu.

Either by using the mouse or by using arrow keys in keyboard, the washer types is selected from the submenu. By loading that four different washer types can be viewed in the screen, either any of one of these can be selected or next screen can be viewed by selecting the "next". Altogether fifteen different washers can be viewed in the screen.

By selecting anyone of the washer, first the file "tha" is loaded. After loading the file, the washer which is selected, first the file "tha" is loaded and for example, if second washer is selected so nw2 program is loaded. In that program, washer is pslit into two parts upper and lower part i.e., outer arc and leg portion forms the upper part and inner arc with two end portions forms the lower part. So, first the lower part is called for, which is stored in seperate program and then the upper part is loaded.

Different inputs are to be given to draw the washer for which that particular washer which is selected is viewed by a slide with different dimensioning parameters. Then, with inputs the washer is drawn on the screen.

Then, the file "DIEDESI" is loaded wherein certain values are being asked and with the inputted values, the program is run. Thickness of washer is first given and the dieplate thickness is searched for, corresponding to the value of thickness. Then, the with previous inputted value, the working area is calcutated and with that, a four pillar dieset is selected which is suitable for the washer needed.

The program is executed and the minimum values of design is listed out in the screen. Then, the fil "DIEDRA" is loaded, wherein taking the minimum values a general punch and dieplate can be drawn. Then, the file "ADD" is loaded, wherein, the cavity corresponding to the type of washer is made. This, is made to rotate through 90 degrees. The entire dieplate with three different views are made as a block. The user defines the name of the block. Then, again the file "DIEDRA" is loaded and thistime for getting the punch drawing. This is also made as a block and is saved with different blockname.

Then, from the software, the user has to quit. Block containing the assembly is inserted and then, punch drawing and diedrawing is inserted in the assembly drawing to get the final assembly for the particular type and size of thrust washer.

# USER'S GUIDE

This software is developed with the view in that even a user with limited knowledge in design and with computer can go in for getting the complete design.

First, the user has to go into the ACAD directory and has to select 1 in the ACAD main menu. Then, after this the file has to be typed in. Then, it goes into the drawing editor. In the command area, MENU should be typed and after the menu name is given. For gettings the part drawings, existing drawings should be selected and in that desired drawing is selected and the part with complete dimensions can be viewed in drawing editor.

If design, is selected , it goes into submenu, wherein, washer is selected. Different types of washer is displayed. Select a particular washer. That washer with different dimensioning variables a slide is projected. Read that and press any key. Thus, it will ask to input the values and do so and automatically load the design procedure.

Input the thickness of the stock, grinding allowance, then, it lists the design values. Get a printout, by connnecting a printer to the computer and press the key printscr. Then after this, it automatically load drawing of punch and dieplate and is saved in two different blocks.

Type insert in command area, it will ask the name of the block. Type assembly and similarly do for punch and dieplate. The assembly is inserted in the drawing editor and then the punch and dieplate is inserted in the upper and lower shoe at respective places. Type "save" and give the file name to save. Necessary dimensioning is done on the assembly and using a plotter, the printouts are taken. Similarly, part drawings are taken out using a plotter.

## PROGRAM LIMITATIONS

This software has certain limitations with reference to maximum and minimum values.

Outer radius can't be more than 127mm

Inner radius can't be more than the outer radius

Outer radius and inner radius can't be zero

Leg angle can't be zero and also can't be more than 180

Height of the washer can't be more than outer radius and can't be less than inner radius

End angle can't be less than zero and can't be equal to or greater than 90

Thickness of the stock can't be zero and greater than 4.25mm, the process can be carried only upto 4.25mm stock thickness, since the available press tonnage is 125 tonnes

## INPUTS

```
Outer radius of Washer     = 63.5 MM
Iners radious of Washer    = 50.8 MM
End angle                  = 45 Degree
Thickness of the washers   = 4.25 MM
Grinding allowance         = 5 MM
```

## OUTPUTS

```
Thickness of Washer        = 4.25 MM
Die plate thickness        = 19 MM
Margin on sides of die opening = 42 MM
Taking production quantity (Million) = Unlimited
Die plate twictness        = 37.5 MM
Width of the die plate     = 147.5 M
length of the die plate    = 211 MM
Area of the die cavity     = 31122.5 MM2
Recommended sizes of screw for fixing the die with die set
      a. Size of screws = M 12
      b. No of screws   = 4
Centre of grantly(y)       = 36.53 MM
Percentage of penetration = 2.38 MM
Press tonnage          = 121.5 tonnes
Length of the punch = 44.5 MM
Clearance between die and punch = 0.35 MM
Stripper plate thickness        = 15.867 MM
```

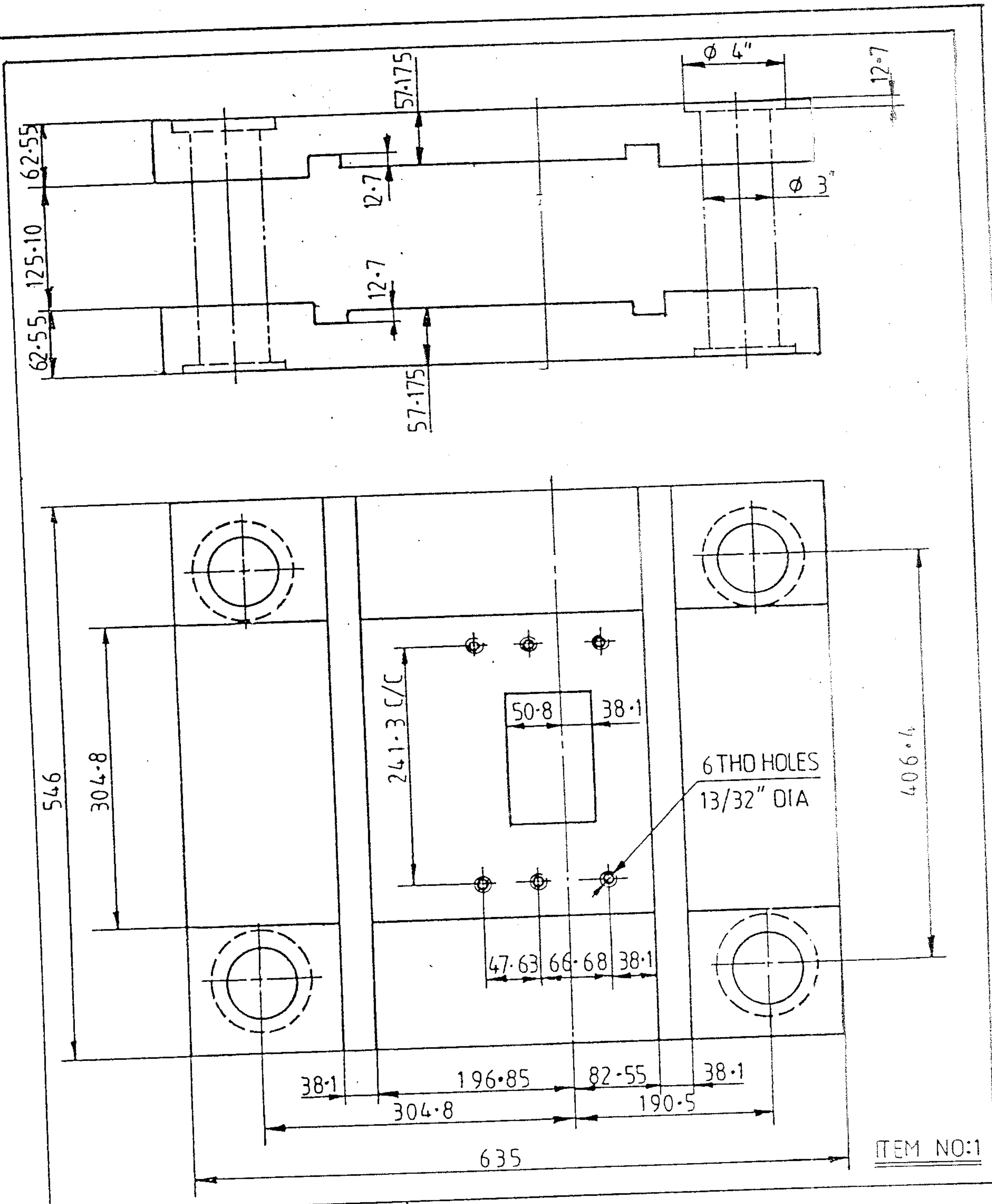
Height of stock strib channel = 7.5 MM

Width of stock strip channel = 139.7 MM

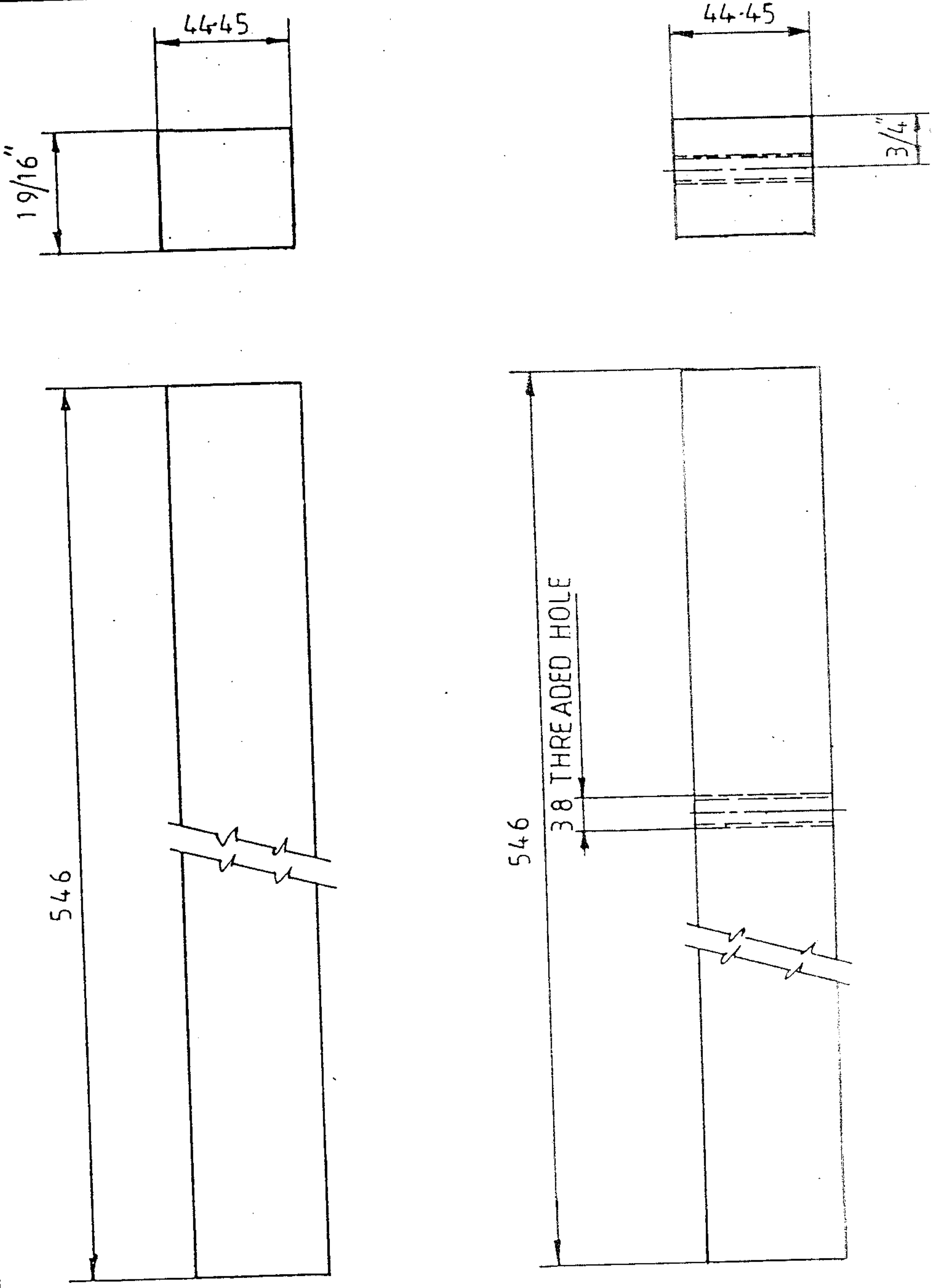
Pitch distace = 54 MM

Multiple pitch distance (taking M=3) = 162 MM

| Working surface axb | A1 | A2 | B1 | B2 | E | D1 | D2 | E1 | E3 | C1 | C2 | B | L | Std theads |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 315x160 | 450 | 440 | 340 | 290 | 350 | 30 | 32 | 200 | 169 | 50 | 65 | 85 | 200 | M50x2 |

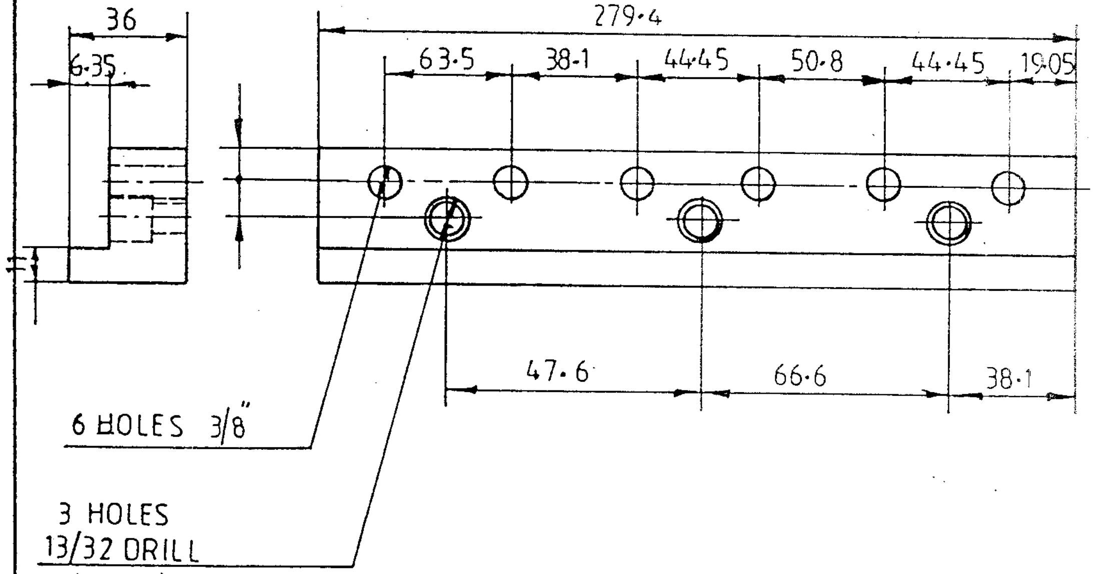


| | |
|---|---|
| DRAWN | KCT |
| CHECKED | DIE SET (With guide pillar) |
| SCALE : 1 : 5 | |
| MATL : HCHcr | |



ITEM NO:2

| | | |
|---|---|---|
| DRAWN | | KCT |
| CHECKED | | GUIDE BLOCK |
| SCALE | 1:2 | |
| MATERIAL | EN8 35RC | |

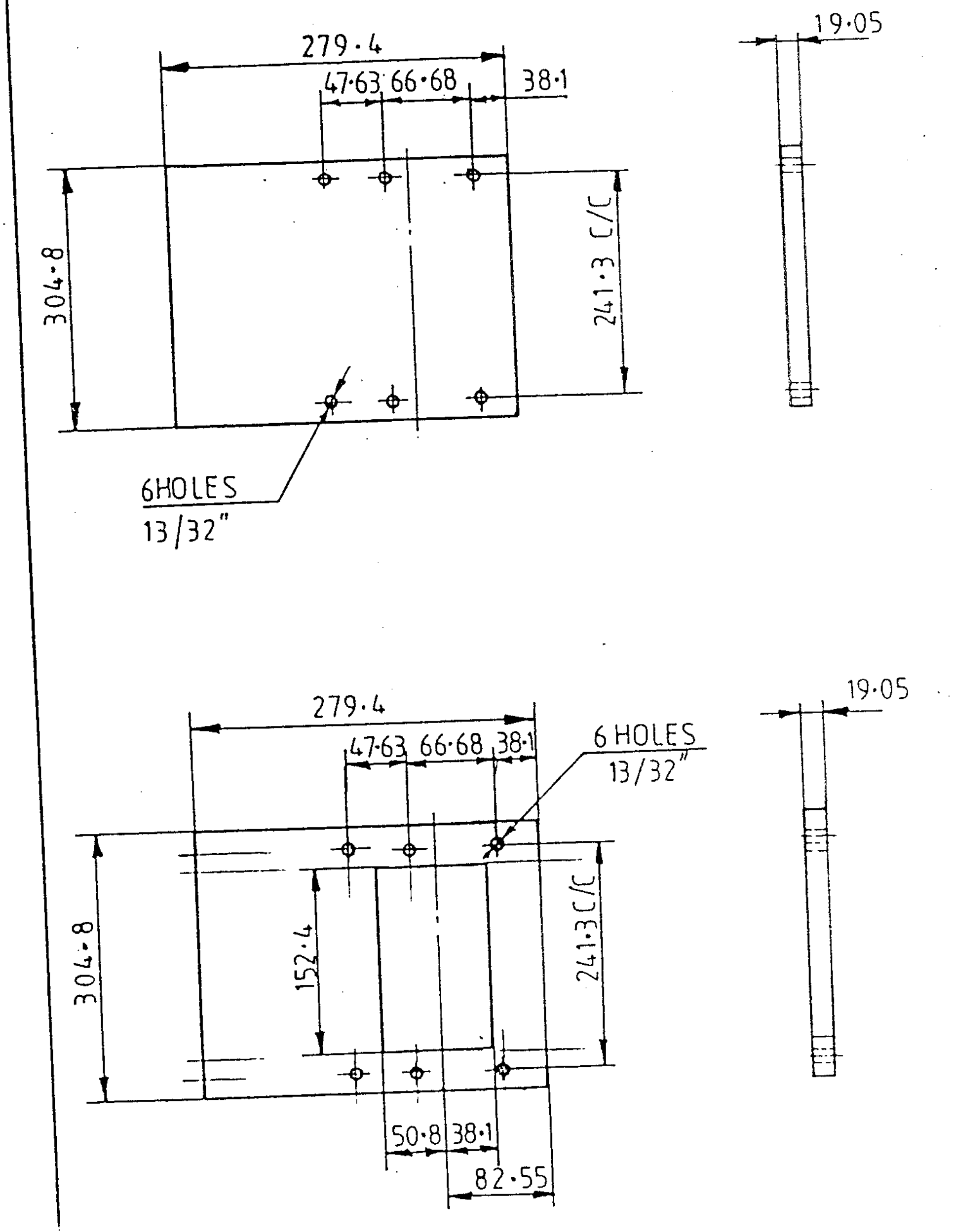


6 HOLES 3/8"

3 HOLES
13/32 DRILL
19/32 C' BORE
5/8 DEEP

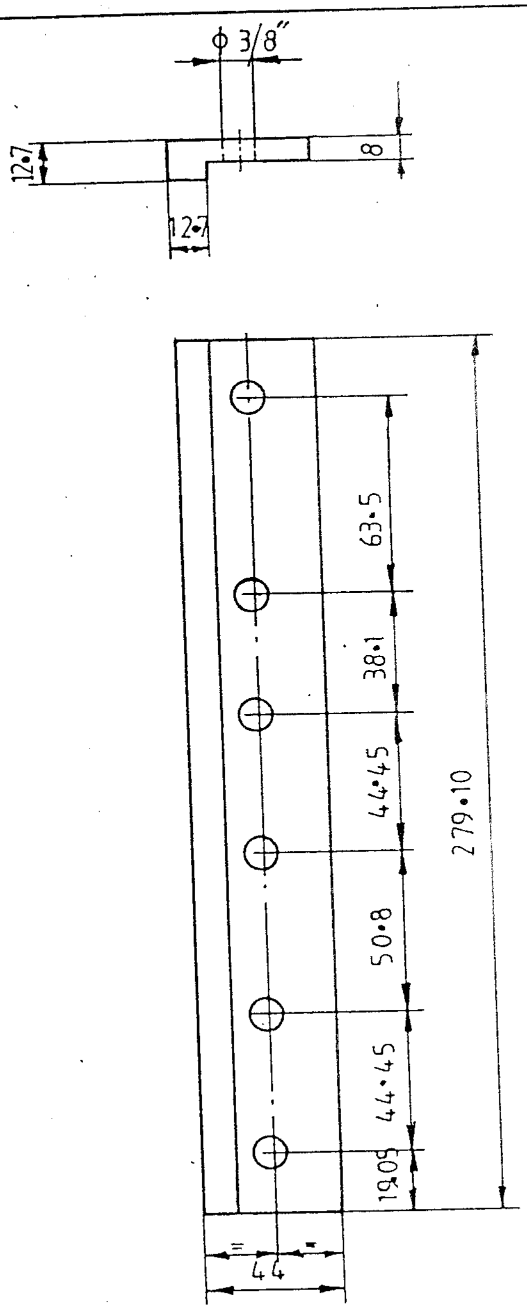
ITEM NO: 3

| | | |
|---|---|---|
| DRAWN | | KCT |
| CHECKED | | SETTING BLOCK |
| SCALE | 1: 2.5 | |
| MATERIAL | EN 8 | |



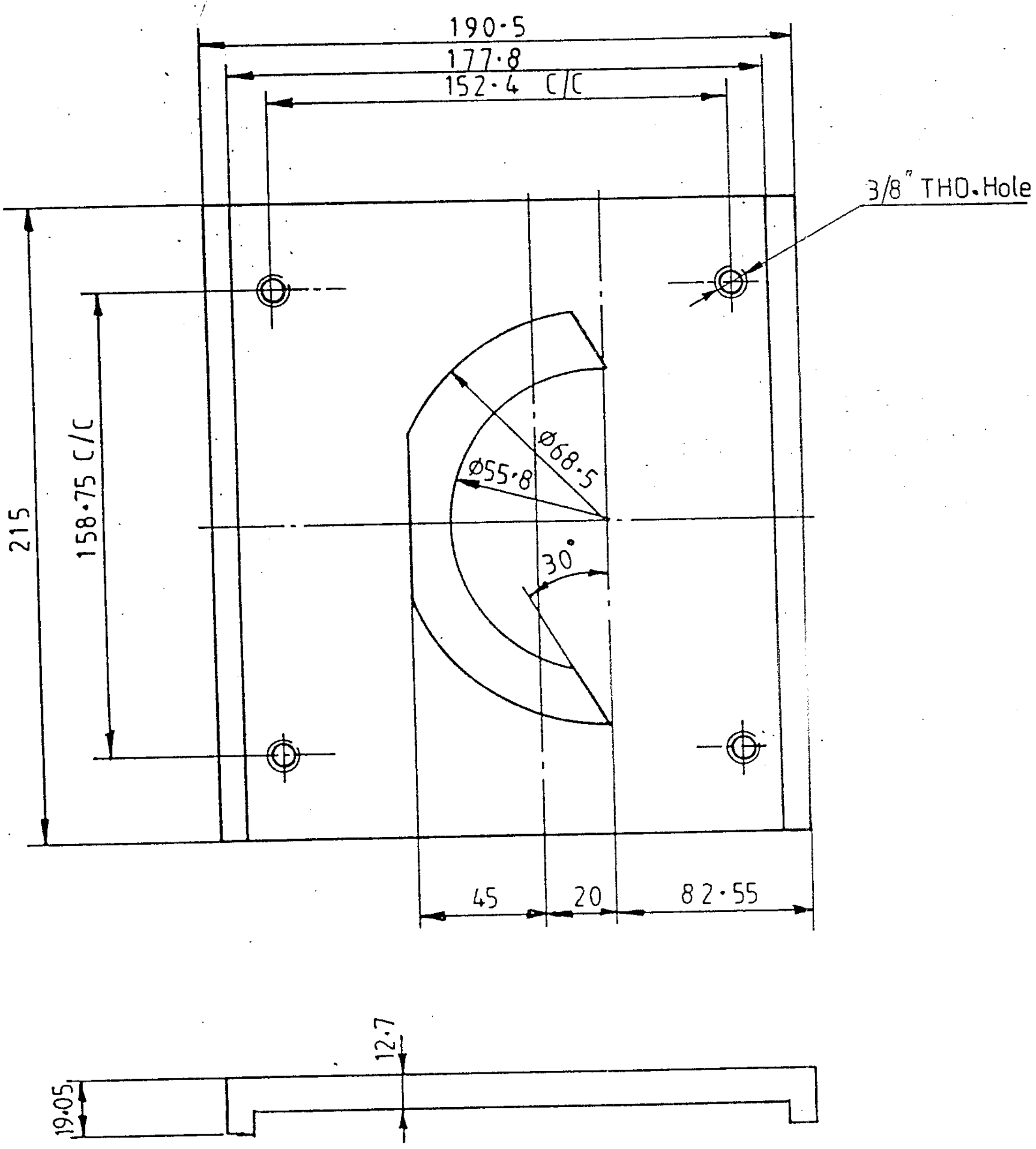
ITEM NO : 4

| | | |
|---|---|---|
| DRAWN | | KCT |
| CHECKED | | HARDENED PLATE |
| SCALE | 1:5 | |
| MATERIAL | HCHCr | |



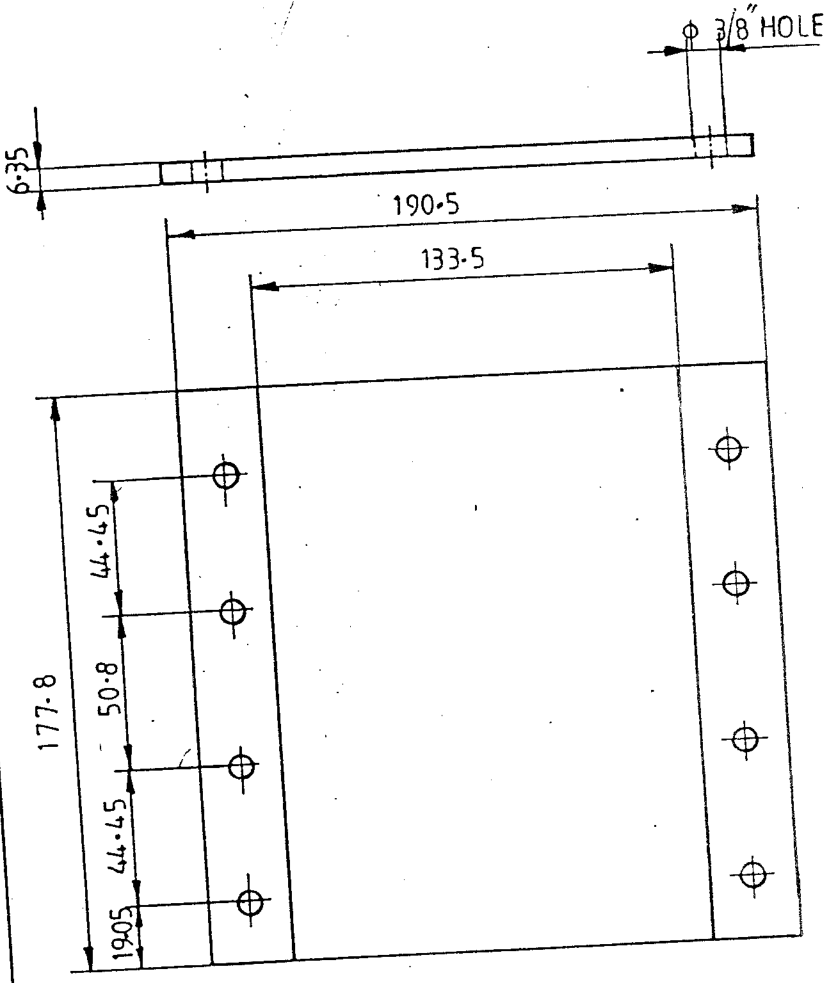
ITEM NO : 6

| | | |
|---|---|---|
| DRAWN | | KCT |
| CHECKED | | CLAMPING PLATE |
| SCALE | 1:2 | |
| MATERIAL. | M.S | |



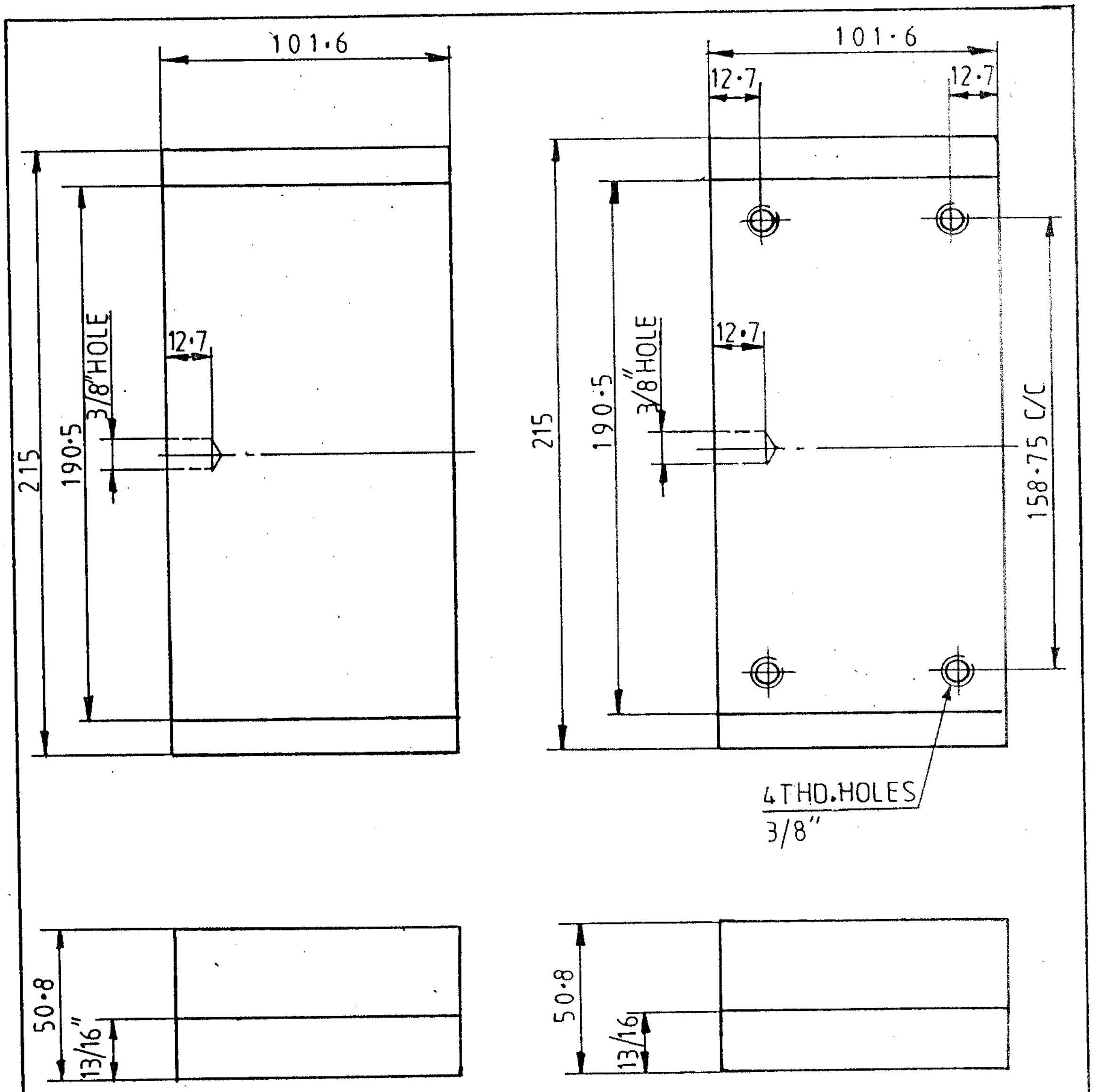
ITEM NO:7

| | |
|---|---|
| DGN | KCT |
| CHD | STRIPPER PLATE |
| SCALE : 1:2 | |
| MATL : M S | |



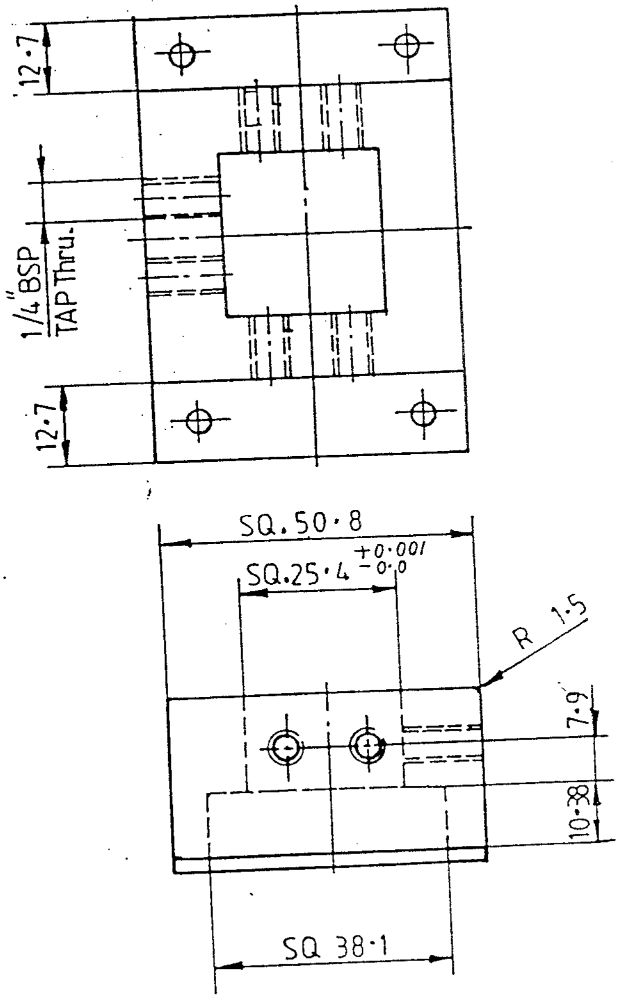
ITEM NO: 8

| | | |
|---|---|---|
| DRAWN | | KCT |
| CHECKED | | GUIDE BLOCK |
| SCALE | 1 : 2 | |
| MATERIAL | M.S | |

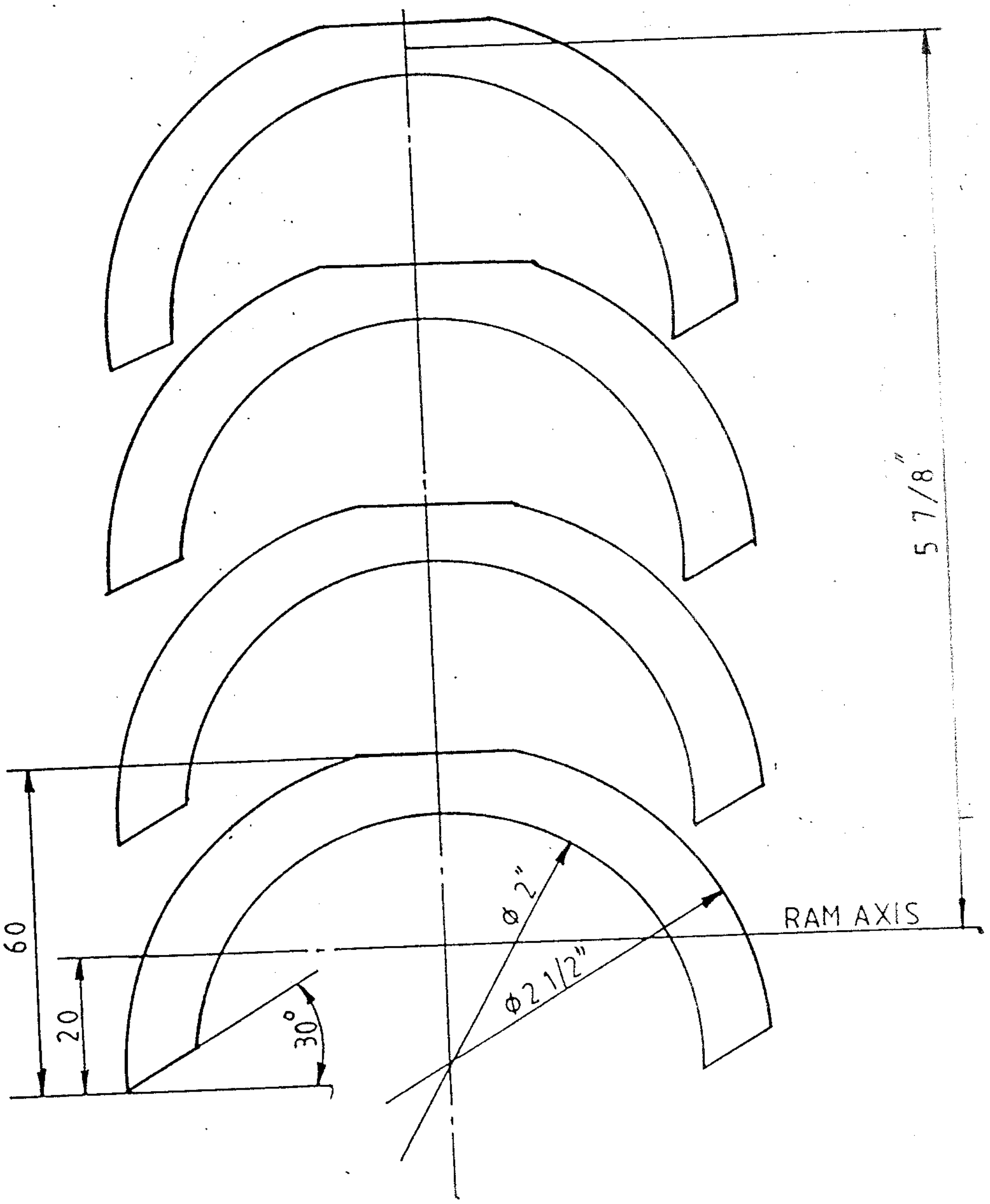


ITEM NO:10

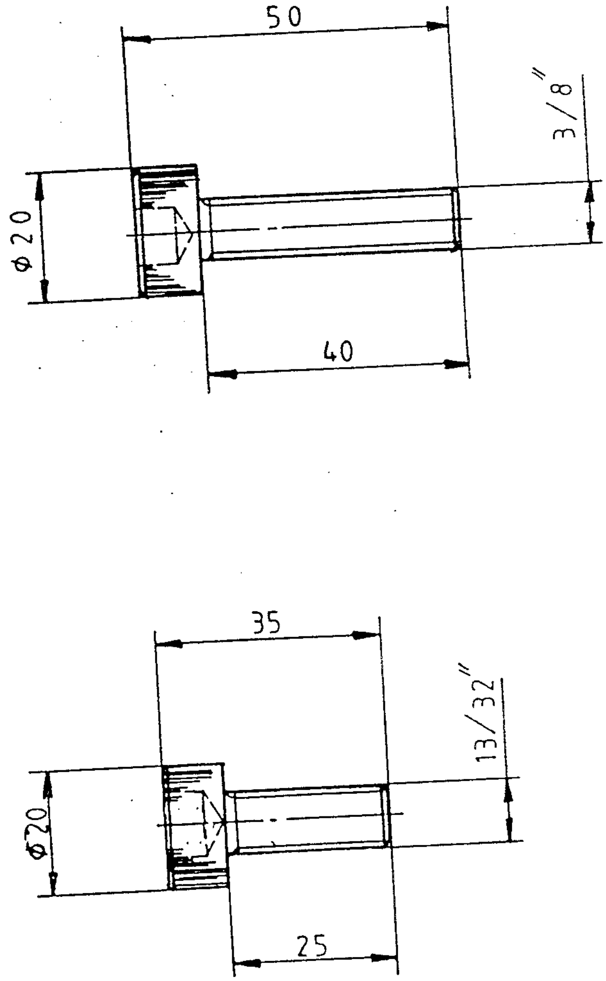
| | |
|---|---|
| DRAWN | KCT |
| CHD | HARDENED PLATE (For stamp holder) |
| SCALE 1:2 | |
| MATL | |



| | |
|---|---|
| DRN | KCT |
| CHD | STAMPING HOLDER |
| SCALE : 1:1 | |
| MATL | |

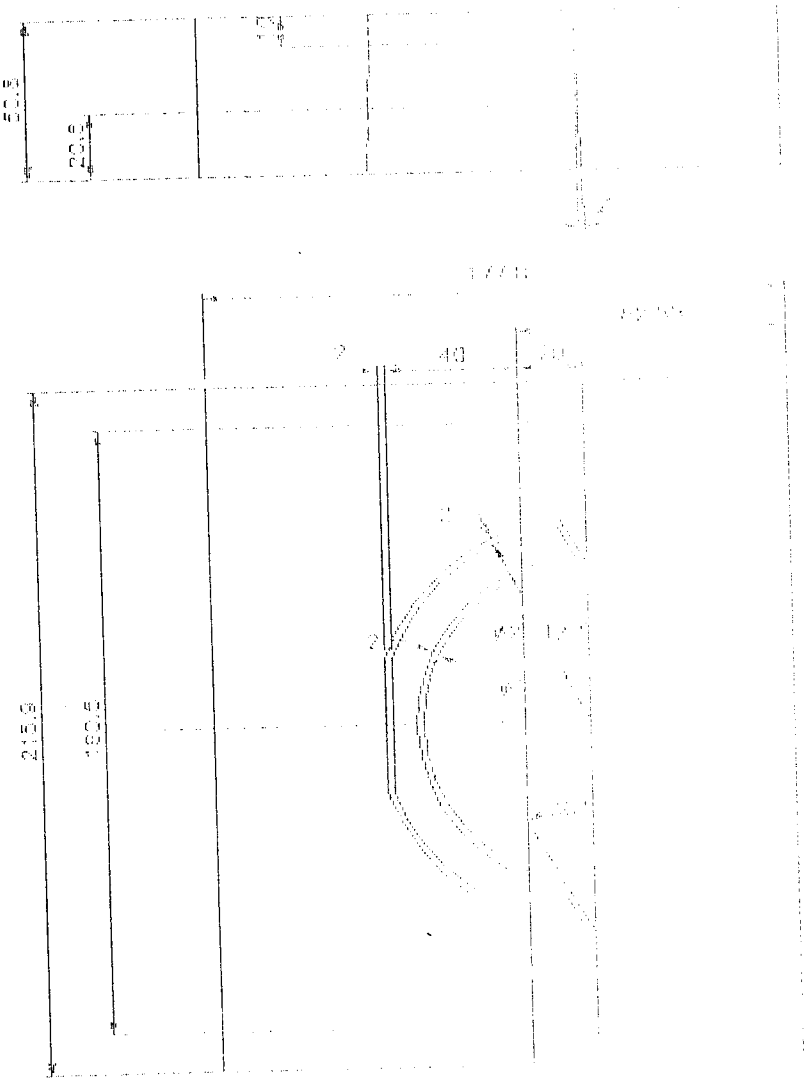


| | |
|---|---|
| DRN | KCT |
| CHD | STRIP LAYOUT |
| SCALE 1:1 | |
| MATL | |



ITEM NO:11

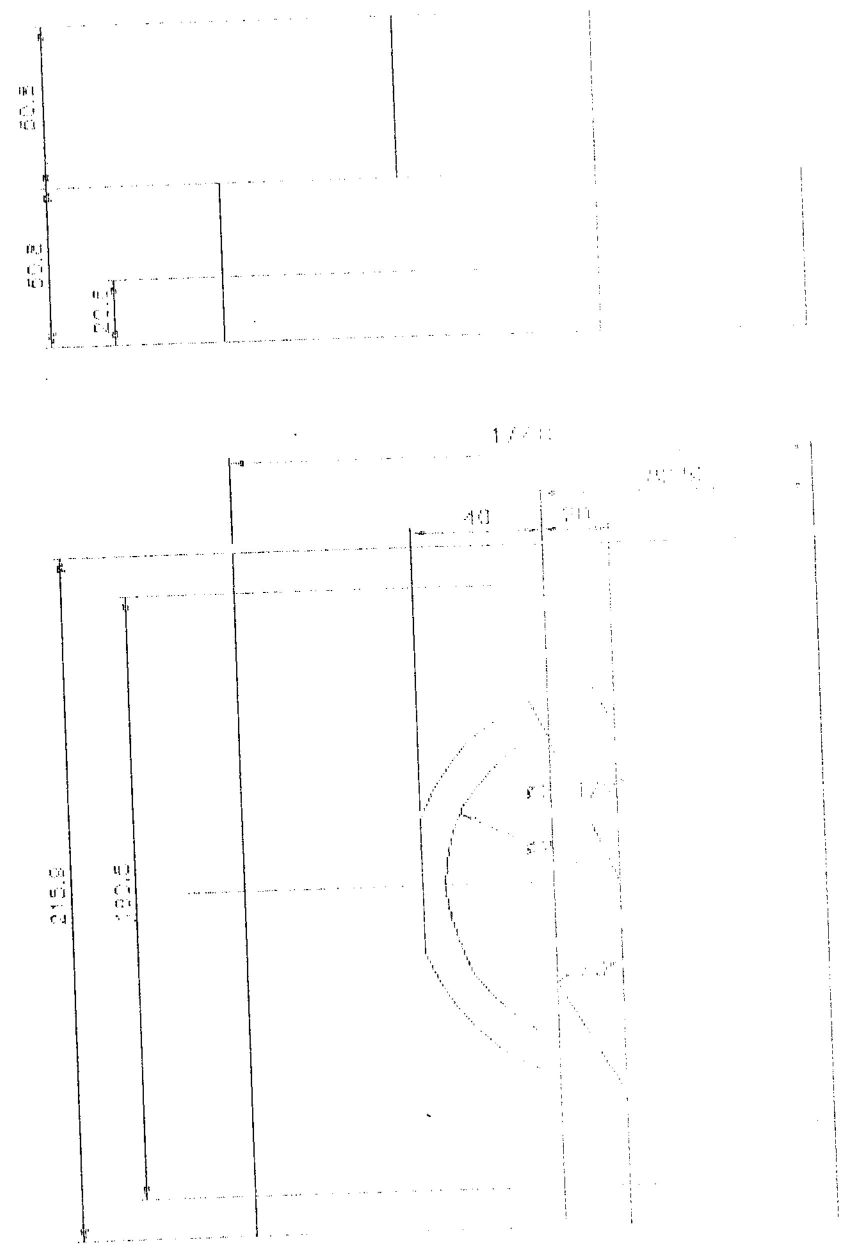
| | |
|---|---|
| DRN | KCT |
| CHD | HEX. SOCKET HD. CAP SCREW |
| SCALE : 1:1 | |
| MATL | |



|  |  |
|---|---|
| DRAWN |  |
| CHECKED |  |
| SCALE | 1:2 |
| MATERIAL | [illegible] |

NUT

DIE BLOCK



|  |  |
|---|---|
| DRAWN |  |
| CHECKED |  |
| SCALE | 1:2 |
| MATERIAL | Holler RC 60 |

KCT

PUNCH

# CONCLUSION

A sincere attempt has been made to resolve the problems faced by the company and initial results prove that problems are more or less solved with this design and is sure it is going to reduce the time for designing by using the software part in future, with the fabricated dieset

# REFERENCE

1. **FUNDAMENTALS OF TOOL DESIGN**

   - AMERICAN SOCIETY OF TOOL & MANUFACTURING ENGINEERS

   PRENTICE HALL PUBLISHERS

2. **PRODUCTION ENGINEERING**

   - P.C. SHARMA

   S. CHAND & COMPANY LIMITED

3. **ALL ABOUT MACHINE TOOLS**

   - GERLING

   WILEY EASTERN LIMITED

4. **INSIDE AUTOCAD**

   - D. RAKER & H. RICE

   BPB PUBLISHERS

5. **THE ABC'S OF AUTOLISP**

   - GEORGE OMURA

   BPB PUBLISHERS

6. **AUTOLISP IN PLAIN ENGLISH**

   - GEORGE O-HEAD

   GALGOTIA PUBLISHERS

7. **ILLUSTRATED AUTOLISP**

- WILLIAM M. OLIVER

  BPB PUBLISHERS

8. **CAD / CAM**

- MIKELL P. GROOVER & EMORY W. ZIMMER JUNIOR

  PRENTICE HALL PUBLISHERS

## AutoCAD COMMAND REFERENCE

Alphabetical list of AutoCAD's commands are given below with brief descriptions and a summary of options. The commands flagged with a prefix in this list are,

**ARC** : Draws an arc of any size. The default method is by specifying two end points and a point along the arc, but several oyher methods are available.

Options :--

A - Included angle.

C - Centre point.

O - Starting direction.

E - End point.

L - Length of chord.

R - Radius.

RETURN - Sets start point and direction as end of last line or arc.

**ARRAY** :

Description :- Makes multiple copies of selected objects in a rectangular or circular pattern.

Options :--

P : Polar (circular) array.

R : Rectangular array.

**AXIS :**

Description :- Displays a "ruler line" on the graphics monitor.

Options :--

| | |
|---|---|
| ON | - Turn axis (ruler line ) on. |
| OFF | - Turn axis off. |
| S | - Lock tick spacing to snap resolution. |
| A(+2) | - Set aspect (differing X-Y spacings) |
| number | - Set tick spacing ( 0 = usnap spacing ) |
| number x | - Set spacing to multiple pf snap spacing. |

**BASE :**

Description :- Specifies orgin for subsequent insertion into another drawing.

**BLOCKS :**

Description :- forms a compound object from a group of entities.

Options :--

? : List names of defined blocks.

**CHANGE :**

Description :- Alters the location, size, orrientation or other properties of selected objects. Especially useful for TEXT entities.

Options :--

P : Change common properties of objects.

C : Color

E(+3) : Elevation.

LA : Layer.

LT : Linetype.

T(+3) : Thickness.

**CIRCLE :**

Description :- Draws a circle of a y size. The default method is by center point and radius but other methods are available.

Options :--

2P : Specify by two end points of diameter.

3P : Specify by 3 points on circumference.

D : To enter diameter instead of radius.

TTR(+2) : Specify by two tangent points and radius.

**COLOR :**

Description :-Establishes the color for subsequently drawn objects.

Options :--

Number : Set entity color number

name : Set entity color to standard color name .

BYBLOCK : Set "floating" entity color.

BYLAYER : Use layer's color for entities.

**COPY :**

Description :- Draws a copy of selected objects.

Options :--

M : Make multiple copies of the selected objects.

**DBLIST :**

Description :- Lists database information for every entity in the drawing.

**DELAY :**

Delays execution of the next command for a specified time used with command scripts.

**DIM** : Invokes dimensioning mode, permitting may dimension notation to be added to a drawing.

**DIM1** : Allows one dimension notation to be added to a drawing, then returns to normal command mode.

**DIST** : Finds distance between two points.

**DIVIDE**: Places workers along a selected object, dividing it into a specified number of equal parts.

**DRAW MODE** : Allows control of the dynamic specification feature for all appropriate commands.

ON - Honor "drag" requests when applicable.

OFF - Ignore "drag" requests.

A - Set "Auto" mode, drag whenever possible.

**DTEXT** : Draws text items dynamically. Here options are same as for TEXT command.

**ELLIPSE** : Draws ellipses using any of several specifications.

C - Specify centre point rather than first axis end point.

R - Specify eccentricity via rotation rather than second axis.

I - Draw isometric circle in current ISOPLANE.

**END** : Exists the drawing editor after saving the updated drawing.

**ERASE** : Erases entities from the drawing.

**FILES** : Performs disk file utility tasks.

**GRID** : Displays a grid of dots, at desired spacing on the screen.

ON - Turn grid on

OFF - Turn grid off

S - Lock grid spacing to snap resolution

A(+2) - Set grid spacing to snap resolution.

Number - Set grid spacing.

Number x - Set spacing to multiple snap spacing.

**HELP OR ?** : Displays a list of valid commands and data entry options or obtains help for a specific command.

**INSERT** : Inserts a copy of a previously drawn part (object) into the current drawing.

Name : Load file "name" as a block.

Name = f : Create Block "name" from file "f"

* Name : Retain individual part entities.

? : List names of defined blocks.

C : (as reply to x scale format) specifies scale via two points (corner specification of scale).

XYZ (+3) : (as reply to x scale prompt). Radius INSERT for x,y and z scales.

**LAYER** : Creats named drawing layers and assigns color and line type properties to those layers.

CC - Set specified layers to color "c"

Fa,b(+3) - Freeze Layers "a" and "b"

? - List layers and their associated colors and the line types.

Lt - Set specified layers to line type "t".

Ma - Make "a" the current layer creating it if necessary.

Na,b - Create new layers "a" and "b".

ON a,b - Turn on layers "a" and "b".

OFF a,b - Turn off layers "a" and "b".

Sa - Set current layer to existing layer "a".

T a,b(+3) - Layers "a" and "b".

**Limits** : Changes the drawing boundaries and controls checking of those boundaries.

2 points set lower left/upper right drawing limits.

ON enables limit checking.

OFF disables limits checking.

**LINE** : Draws straight lines of any length.

RETURN (as reply to "FROM POINT") start at end of previous line or Arc.

C (as reply to "TO POINT") close Polygon.

U (as reply to "TO POINT") undo Segment.

**LINE TYPE** : Defines line types (sequencing of alternating line segments and spaces), loads them from libraries, and sets the line type for subsequently drawn objects.

? - List a line type library

C - Create a line type definition.

L - Load a line type definition.

S - Set current entity line type.

"set suboptions"

name - Set entity line type name.

BYBLOCK - Set "floating" entity line type.

BY LAYER - Use layer's line type for entities.

? - List loaded line types.

**LIST** : Lists databases information for selected objects.

**LOAD** : Loads a file of User-defined shapes to be used with the SHAPE command.

? - List the names of loaded shape files.

**LTSCALE** : Sets scale factor to be applied to all linetypes within the drawing.

**MENU** : Loads a file of drawing editor commands into the menu areas (Screen,Pull-down,Tablet and button).

**MINSERT** : Inserts multiple copies of a block in a rectangular pattern.

name - Load file "name" from a rectangular array of the resulting box.

name-f - Create Block "name" from file "f" and form a rectangular array.

? - List names of defined blocks.

C - (as reply to X-scale prompt) specifies scale via two points (concern specification of scale)

XYZ(+3) - (as reply to x scale prompt) readies MINSERT for X,Y and Z scales.

**MIRROR** : Reflects designated entities about a user specified axis.

**MOVE** : Moves a designated entities to another location.

**MSLIDE** : Makes a slide file from the current display.

**MULTIPLE** : Causes the next command to repeat until cancelled.

**OFFSET** : Allows the creation of offset curves and parallel lines.

number - specifies offset distance.

T - Thorough : Allows specification of a point through which the offset curve is to pass.

**PLINE** : Draws connected line and arc segments with optional width and taper.

H - Set new half-width.

U - Undo previous segment.

W - Set new line width.

RETURN - Exit PLINE command.

In line mode

A - Switch to arc mode.

C - Close with straight segment.

L - Segment length (continoue previous segment).

In arc mode

A - Included angle.

CE - Centre point.

CL - Close with arc segment.

D - Starting direction.

l - Chord length or switch to line mode.

R - Radius.

S - Second point of three-point arc.

**QTEXT** : Enables text entities to be identified without drawing the text detail.

ON - Quick text mode on.

OFF - Quick text mode off.

**QUIT** : Exits the drawing editor and returns to AuotoCAD's main menu discarding any changes to the drawing.

**REDRAW** : Refreshes or cleans up the display.

**ROTATE** : Rotates existing objects.

R - Rotate with respect to reference angle.

**SETVAR** : Allows you to display or change the value of the system variables.

**SHELL** : Allows access to other programs while running AutoCAD.

**UNDO** : Reverses the effect of multiple commands, and provides control over the "undo" facility.

number undues the number most recent commands.

A Auto - controls treatment of menu items as undo groups.

B back - Undoes back to previous undo mark.

C control - Enables/Disables the UNDO feature.

E End - Terminates an UNDO group.

G Group - Begins sequence to be treated as one command.

M Mark - Places marker in UNDO file (for Back).

**VSLIDE** : Displays a previosly-created slide file.

File - View slide *File - Pre-loaded slide, next VSLIDE will view.

**WBLOCK** : Writes selected entities to a disk file.

name - Write specified block definition.

= - Block name same as file name.

* - Entire drawing.

Return - Write selected objects.

**Zoom** : Enlarges or reduces the display of the drawing.

number - Multiplier from orginal scale.

number x - Multiplier from current scale.

A - All.

C - Center.

D(+3) - Dynamic Pan/Zoom.

E - Extents ("drawing uses").

L - Lower left corner.

P - Previous.

W - Window.

# AUTOLISP COMMAND REFERENCE

**ACAD.LSP** : AutoLISP file loaded each time you enter the drawing editor.

**ACAD.PGP** : File available with AutoCAD to define programs accessible from within the drawing editor.

**ACAD FREERAM** : Adjustable area of working memory used by AutoCAD should be set to 24.

**angle** : AutoLISP command that measures the angle of the known points (angle pnt1 pnt2)

**angtos** : AutoLISP command that converts a variable stored as radians to a string in another format (set of a (angtosang 1)).

**Apostrophe** : Used to indicate a list (set a (3 57))

**Argument** : A value passed from outside a program to a variable within an AutoLISP program.

**assoc** : The AutoLISP command that searches for a sublist within any list. With an entity list, assoc uses the entity code number askey(setq c(assoc 40 b)).

**Atom** : A single element.

**AUTOEXEC.BAT** : A batch file executed when the computer is first booted.

**BASIC** : A high level language often used to program microcomputers.

**Breakpoint** : A stop point placed in a program for the purpose of debugging. .pa

**Caddr** : AutoLISP command that produces the third element in the list.

**cadr** : AutoLISP command that produces the second element in the list.

**car** : AutoLISP command that produces the first element in the list.

**cdr** : AutoLISP command that produces the second and remaining elements in a list.

**command** : AutoLISP command that lets you use AutoCAD commands the AutoLISP programs [ command "Line" Pnt1 Pnt2 "]

**crossing** : To create a window in AutoCAD in order to select the entities. If any part of an entity touches the window, it is selected.

**debug** : To correct the program logic and syntax errors.

**defun** : The first command in an AutoLISP program ; defines the name of the function or program.

**distance** : AutoLISP command that measures the distance between two points. (setq a (distance pnt1 pnt2)).

**divide** : Arithmetic operator (/ a 3) means "a" divide 3.

**dtr** : In AutoCAD, an AutoLISP command that converts the value of angles from degrees to radians.

**edit** : AutoCAD command for entering EDLIN text editor from inside AutoCAD, defined in ACAD.pgp file.

**element** : A single value in a list.

**else** : The third statement in the if-then-else statement [ of (= a 5) (set 9 b 6) (set 9 b 7) ] .pa

**entget** : AutoLISP command that secures an entity list (set 9 b (entget na) ]

**entity** : The smallest element that may be placed in a drawing with a single command. Typical entities are ARC LINE CIRCLE POINT e.t.c

**entsel** : AUTOLISP command that selects an entity for VIEW and CHANGE. Only one entity may be selected with this command. (setq a (entsel) ]

**eqaul (=)** : Autolisp command used only to test quality. It isn't an assignment command (if (=a5)

**Explanation** : AutoLISP shorthand command that prints the value of a variable

**Extension** : AutoLISP The second half of a DOS filename. The required extension for all autoLISP files is .LSP.LES1.LSP

**Findfile** : searches for a requested file. If the file is found, then the complete directory path is returned. If the file is not found then nil is returned.

**Function** : An AutoLISP program.

**getangle** : AutoLISp command that lets you find an angle by pointing to two points or by entering the angle from the keyboard (setq a (getangle)]

**getcorner** : AutoLISP command that asks you to pick a second point on the screen with the mouse or cursor. A window is draaged visually across the screen from point t the second point (setq Pnt 2 (get corner pnt 1 "Pick second point "0 ]

**getdist** : AutoLISP command that requests input as a real number either through the keyboard or by pointing .pa

**getkword** : AutoLISP command used instead of getsring to verify only selected inputs. Used with (initge-1) (initget-1"Ẏ"N) (seta (getword "Y or "N )

**getpoint** : AutoLISP command that asks tou to specify a point on the screen [(setq a (getpoint "pick a point ")]].

**getreal** : AutoLISP command that requests keyboard input as a real number.

**getstring** : AutoLISP command that requests keynoard input as a string.

**getvar** : AutoLISP command that produces an AUTOcad system variable [setq a (getvar "osmode" 0)]

**global** a variable whose value remains when the program ends and is usable by other programs.

**graphics** : AutoLISP command that shifts the monitor to the drawing editor.

**greater than** : ( ) Basic comparitive (if ( a b ) reads if 'a' is greater than 'b'.

**less than** : ( ) Basic comparative (if ( a b )] reads if 'a' is less than 'b'.

**Lisp** : High level language used currently in artificial intelligence and used as the basics for AutoLISP.

**List** : A variable with more than one value.

**Load** : Used to load AutoLISP program files into memory so that functions can be executed (load "les1").

**Local** : A variable whose value is available to one program.

**nil** : No value assigned to a variable. .pa

**or** : Basic connector generally used to connect items of comparison in 'if' statements.(if (or ( = b a ) ( a c ))) reads if b = a or 'a' is greater than 'c'.

**Paranthesis()** : Each AutoLISP command and list is surrounded by paranthesis.

**Pick** : Refers to choosing a point or object on the screen.

**Polar** : AutoLISP command that derives a point at a given distance and angle from another known point (setq pnt2(polar pnt1 ang1 dist1)).

**Princ** : AutoLISP print command.

**Print** : AutoLISP print command that adds a line feed.

**Progn** : AutoLISP command that groups several AutoLISP statements into one for use as a single then or else statement. (if(= a b) (Progn (xxx)(xxx)(xxx))).

**Prompt** : AutoLISP command that prints to the screen.

**Quatation "** : Generally encloses strings when assigned to variables or used as constants.

**repeat** : AutoLISP command that repeats a loop x number of times (repeat 5).

**Return** : AutoLISP produces a value when a command issued, often referred to as returning a value.

**semicolon ;** : Used to denote a non-executable comment line in AutoLISP.

**setq** : AutoLISP's basic assignment command (setq a b) assigns the value of 'b' to 'a'.

**setvar** ; AutoLISP command that sets AutoCAD system variables.(setvar "OSMODE" 1).

**slash /** : The slash before variable variable names in the (defun command denotes local variables).

**strcase** : AutoLISP command tjat evaluates all characters in a variable as uppercase.(setq a (strcase b)).

**string** : A group of alphanumeric characters enclosed in quotes.

**subst** : AutoLISP command that substitutes one entity sublist for another with an entity list.(setq b1(subst(40.02500000) cb)).

**subtract** : Arithmetic operator (-a1)means subtract 1 from a.

**symbols** : same as variables.

**system variables** : selectable variables that control certain AutoCAD defaults.

**Text Editor** : Program used to produce and edit text files Norton Editor is an example of a text editor.

**Text file** : ASCII file without control codes or other special coding sequences used by word processors.

**Variable** : A combination of letters or letters and numbers used to store other values.

**While** : One of these basic loop statements in AutoLISP. the loop will continue while the variable or expression is not 'nil'.

**Windowing** : The process of creating a window in AutoCAD to select entities. Only full entities within the window are selected.