COMPUTER AIDED GEOMETRIC DESIGN OF FORM-ROLLS FOR NC MANUFACTURE

by

KOON HUEK NG

A Thesis submitted as Partial Requirement
for the Degree of

DOCTOR OF PHILOSOPHY

to

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Department of Production Technology and
Production Management

November 1981

THE UNIVERSITY OF ASTON IN BIRMINGHAM

COMPUTER AIDED GEOMETRIC DESIGN OF FORM-ROLLS FOR NC MANUFACTURE

KOON HUEK NG BSc (Hons)

(A Thesis submitted for the PhD Degree, 1981)

SUMMARY

Conventional methods of form-roll design and manufacture for Cold Roll-Forming of thin-walled metal sections have been entirely manual, time consuming and prone to errors, resulting in inefficiency and high production costs. With the use of computers, lead time can be significantly improved, particularly for those aspects involving routine but tedious human decisions and actions. This thesis describes the development of computer aided tools for producing form-roll designs for NC manufacture in the CAD/CAM environment. The work was undertaken to modernise the existing activity of a company manufacturing thin-walled sections.

The investigated areas of the activity, including the design and drafting of the finished section, the flower patterns, the 10 to 1 templates, and the rolls complete with pinch-difference surfaces, side-rolls and extension-contours, have been successfully computerised by software development. Data generated by the developed software can be further processed for roll manufacturing using NC lathes.

The software has been specially designed for portability to facilitate its implementation on different computers. The Opening-Radii method of forming was introduced as a subsitute to the conventional method for better forming. Most of the essential aspects in roll design have been successfully incorporated in the software.

With computerisation, extensive standardisation in existing roll design practices and the use of more reliable and scientifically-based methods have been achieved. Satisfactory and beneficial results have also been obtained by the company in using the software through a terminal linked to the University by a GPO line. Both lead time and productivity in roll design and manufacture have been significantly improved. It is therefore concluded that computerisation in the design of form-rolls for automation by software development is viable. The work also demonstrated the promising nature of the CAD/CAM approach.

Keywords: CAD/CAM
COLD ROLL-FORMING
DESIGN OF FORM-ROLLS
SOFTWARE DEVELOPMENT AND APPLICATIONS
NC MANUFACTURE

A C K N O W L E D G E M E N T S

I am deeply grateful to the following for the successful completion of the work presented in this thesis:

Dr D A Milner, my supervisor, for his guidance, advice and encouragement.

Professor R H Thornley, Head of the Department of Production Technology and Production Management, for his continuing support and interests in the work.

The University of Aston in Birmingham, for financially supporting me throughout the course with a studentship grant.

Mr Geoff Deeley, Works Director of the company, and his colleagues, for their consistent interests in the work and their assistance in the investigations.

The staff of the University Computer Centre, for their help in the use of the computer facilities.

Donna and Jean for typing the thesis.

Last but not least, my mum and Poh Kheng, for their affection and encouragement.

DECLARATION

No part of the work described in this thesis has been submitted in support of an application for another degree or qualification of this or any other University or other institution of learning.

C O N T E N T S

Page

Page

LIST OF FIGURES, PLATES, TABLES AND CHARTS

Page

Figures (Cont/d)

Page

Figures (Cont/d)

Page

Figures (Cont/d)

Page

Figures (Cont/d)

Plates

Page

Plates (Cont/d)

Tables

Page

Tables (Cont/d)

Page

Tables (Cont/d)

Charts

Page

Charts (Cont/d)

CHAPTER 1

INTRODUCTION

The advent of programmable digital computers can be considered a great stride forward in the history of mankind. With this new technology comes a decisive and persistent wave of revolutionary changes which are to last for generations to come, eventually transforming human life style and activity in almost any conceivable way. To date, computer technology has already helped to shoulder much of the human burden in providing the much needed material comforts and conveniences in modern society, notably in the industrial, commercial and educational sectors. Until more superior technology emerges, the influence of computers is most likely to deepen with time as mankind increasingly relies on them for greater material prosperity.

Computer-Aided Design (CAD) and Computer-Aided Manufacture (CAM), among many others, are two particularly booming areas of computer application in engineering. Much effort had long been spent in exploring the mathematical domain for suitable theories and means of representing and generating geometrical shapes in the form of curves and surfaces from which computerised design

methods for specific applications may be developed. Typical pioneers in this field such as Coons[1], Forrest[2], Bezier[3] and many others had helped to firmly establish the rightful role of CAD in future. CAM on the other hand has been developed chiefly out of the needs for industrial automation in the manufacturing industries and the invention of Numerical Control (NC) machines can be regarded as the major breakthrough in this respect. The progress from Conventional Hard-wired NC to Computer Numerical Control (CNC), Direct Numerical Control (DNC) and now CAM has been steady and rapid. The days of totally unmanned factories supervised by computers are in fact not very far off. The entire future commitment of computers in engineering design and production, as well as in many other spheres, is currently being enthusiastically pursued on an international scale.

Current state of CAD/CAM progress in the United Kingdom is reflected in some detail in a report by Leslie[4] on the response of the Department of Industry to recommendations on CAD/CAM published by the Advisory Council for Applied Research and Development (ACARD). Bodies exclusively funded by the Department of Industry in these fields include the Computer-Aided Design Centre (CADC) at Cambridge and the National Engineering

Laboratory (NEL). Informal interests in these fields are individually pursued by industry linked bodies, including research associations like Production Engineering Research Association (PERA) and Machine Tool Industry Research Association (MTIRA), educational establishments like universities and polytechnics, various professional institutions and so on, quite frequently with support from organisations like the Science Research Council (SRC). Much still needs to be done to promote general awareness of the use of CAD/CAM, as well as to develop CAD/CAM expertise in yet unexplored fields.

The Department of Production Technology and Production Management at the University of Aston in Birmingham, has long been fostering close links with industrial organisations in the application of high technology to further industrial progress. In response to the needs of a leading West Midlands company in the sheet-metal section manufacturing industry, the Department has taken the initiative to develop in collaboration with the company, computer-aided design and manufacturing facilities vital to the future of the company.

This research has been planned and implemented, under the present CAD/CAM perspective, to fulfil an important

part of the company's objective in modernising its current activity. The subject of sole concern here is computerisation of the form-roll design and manufacturing activity, which is the key activity of Cold Roll-Forming used by the company. The main aspects of the work involved in this research are:

1. Investigation of relevant areas of the existing activity in the company related to the design and manufacture of form-rolls.

2. To select from the investigated areas suitable and beneficial parts of the form-roll design and manufacturing methods for computerisation.

3. To devise a research programme which could benefit the company in both short-term and long-term periods without adversely affecting current company design and manufacturing interests.

4. To implement the research programme through stage by stage software design and development, keeping continuous liason with the company at all stages.

CHAPTER 2

COLD ROLL-FORMING PROCESS

2.1 Definition of Cold Roll-Forming

The definition of Cold Roll-Forming (Plate 2.1) used by the Cold Rolled Sections Association[5] is as follows:

Cold Roll-Forming is a process of forming metal from strip or sheet of uniform thickness, into shapes of essentially constant cross section, generally by feeding the material longitudinally through successive pairs of rolls progressively forming the metal until the finished cross section is achieved.

Similar definitions have also been used by other related organisations, such as the American Society of Manufacturing Engineers[6], the American Society of Mechanical Engineers[7] and the American Society for Metals[8]. The process is also known as Contour Roll Forming.

2.2 Machinery and Tooling

A typical machine (Plate 2.2) used for Cold Roll-Forming, generally consists of a horizontal steel table

of a fixed width and a number of roll-forming stations mounted on it in succession. The capacity of a roll-forming machine is usually limited by its table width, the maximum number of roll-forming stations it carries and the size of the stations. Large roll-forming stations are usually required to accommodate rolls of large diameter in forming large sections of thicker materials. However, most cold rolled sections are manufactured from thin materials, hence standard machines which carry relatively small stations are more frequently used. The maximum number of stations carried by these standard machines may range from about ten to as many as twenty-five. A machine which carries a greater maximum number of stations normally has a greater table width as well.

Each roll-forming station (Plate 2.3) contains a pair of horizontal rolls, which are carried by two horizontal shafts or spindles supported by mountings on both sides of the machine table. Occasionally, the use of short spindles in the forming of narrow sections may require only mounting on one side of the table for support. One or two vertical rolls may be included at the same station to perform part of the forming operation. They are usually referred to as side-rolls and are blended to the top and bottom rolls, either on one side

or on both sides. At each station, the distance between the mountings across the table can be adjusted and so are the heights of top and bottom rolls. For the latter micrometer facilities are normally available. The rolls of each station are usually power driven through a system of gears and have their peripheral speeds synchronised. The speed of forming or the speed of material travel may vary from station to station and usually that too requires synchronisation. Roll surfaces are usually lubricated to reduce friction between roll surface and material surface so as to maintain a good surface finish of the material and also to cool both the rolls and the material which may expand as a result of heat generated during forming.

To aid forming operations, accessories may sometimes be used. Strip guides (Plate 2.4) are normally used before and in between the roll passes to keep the material in the right track. Inter-station idler-rolls (Fig 2.1) may be used for the similar reason and in addition for doing partial forming on the material as well. A straightening-device (Fig 2.2) may be placed immediately after the last roll-forming station to straighten the section. If coiled stock is used for forming while products of fixed length are required, then an additional cutting-off operation with a special machine

(Plate 2.5) will be included at the end of the entire forming process.

2.3 Forming Operation

When the strip is passed through the rolls at each forming station, its cross-sectional shape tends to assume the guiding contours of the rolls progressively until forming is completed (Plate 2.6). During forming, non-linear parts of the strip are bent by the rolling action while linear parts normally remain unchanged. To fix the shape of the finished section, an operation called coining or ironing is used and it is accomplished by including a series of roll stations which carry rolls of contours identical to the finished section at the end of the entire forming process.

2.4 Auxiliary Operations

Auxiliary operations which can be incorporated in Cold Roll-Forming include the usual press-tool operations like piercing, blanking and notching, sweeping the finished section into circular radius if curved sections are required, seam welding for forming closed tubes and so on.

2.5 Form-Roll Design and Manufacture

In general, the production of each new section would require an entirely different set of rolls to be designed and manufactured. The reasons are that each set of rolls is only useful for producing the section shape it is intended and that in practice, except for a few standardised products, a section is rarely manufactured in repeated batches. Roll design and manufacture generally undergo the following common stages, though individual approaches may be different.

2.5.1 Determination of Strip Size

By making some reasonably valid assumptions about the probable behaviour of strip material during forming and about the ultimate shape of a section as a result of forming, approximate width of the raw material or strip size can be derived. The estimated strip size will then be used in both roll design and other evaluations.

2.5.2 Design of Forming Sequence

This stage usually involves the undertaking of all design considerations and calculations necessary to establish both the number of roll stations to be used

and the whole sequence of transitional shapes of the section to be formed station by station. Basically, designers have to determine the angles and radii for successive bending of each non-linear part in a proper manner so as to ensure satisfactory forming performance during production. All aspects of the expected material behaviour in the given situations at each stage of forming must be examined and every appropriate design action must be taken to control the material movement so that disastrous results can be avoided. Typical material behaviour like spring back, thinning, stretching and persistence of flow have to be carefully considered. To keep down the cost of roll design and manufacture, normally only the minimum number of roll stations required are used unless the production quantity is large enough to justify the use of more stations for prolonged roll life with less severe bending at each station.

2.5.3 Design of Rolls and Accessories

At this stage, a suitable combination of tooling at each station has to be determined for the required sequence of forming. Decisions have to be taken about things like whether to use whole rolls or split-rolls, whether to use side-rolls, idler-rolls or just guides,

the number of ironing stations and so on. Suitable allowance and adjustments to the roll contours must also be incorporated for smooth forming. Having finalised the roll design, a set of drawings is then produced for roll manufacturing purposes.

2.5.4 Manufacture of Rolls

There are different but equally valid approaches regarding how the rolls should be manufactured. Some establishments may prefer the traditional method of producing detailed and dimensioned drawings for subsequent machining, while others may choose to avoid that by adopting accurate copying techniques, like the use of master templates or some form of contour tracing facilities. Despite the differences, rolls are normally machined on lathes of some kind. After machining, a test run is usually conducted for the rolls before they are sent for surface hardening and grinding. Once set up for production, the rolls are seldom replaced unless some unusual faults develop.

2.6 Materials for Cold Roll-Forming

Materials suitable for Cold Roll-Forming are normally any kind of metal which have sufficient ductility and which do not fail easily when work-hardened without

annealing. They include low and medium carbon steels, alloy steels, stainless steels, aluminium and its alloys, copper, brass, phosphor-bronze and zinc. Occasionally, even titanium, nickel-base heat-resisting alloys and high strength steels may also be formed. There are British Standards available for materials commonly associated with Cold Roll-Forming[9 - 16]. Normally, metal as thin as 0.005 in. and up to 0.75 in. of thickness can be satisfactorily formed. The raw materials used may come in the form of coiled stock or flat strips or sheets. Pre-coated materials may also be formed satisfactorily without impairing their surface finish, provided the coating is sufficiently elastic. Typical examples are those with zinc-coating, aluminising-coating, plastic-coating and paint-coating.

2.7 Cold Roll-Forming Products and their Applications

Cold Roll-Forming products are sections made from all combinations of suitable material, thickness, length, coating or surface finish, shape and size. They are mainly used in applications for structural purposes or aesthetic purposes or both. The main types of application include building components such as purlins, rails and channels, automobile components such as chassis and bumpers, frames for cargo containers, consumer goods like office equipment, furniture and domestic appliances,

shop fittings, shelving and racking, and also aircraft and shipbuilding components.

2.8 Advantages of Cold Roll-Forming and its Products

The widely recognised advantages are as follows:

(a) Profile of rolled sections are consistent and accurate.

(b) Virtually any profile can be produced by the process.

(c) Wide range of ferrous and non-ferrous metals can be readily formed by the process.

(d) Pre-coated materials can be formed satisfactorily by the process.

(e) The sections can be produced to exact lengths thus reducing wastage.

(f) High strength to weight ratio of the sections, especially thin-walled sections, means greater value and less expenses.

2.9 Rival Processes

On the cold forming side, only the press-brake process constitutes a significant challenge to Cold Roll-Forming. The press-brake process is not as versatile in terms of the section shapes it can cope with. It requires very bulky machinery and is usually suited for low quantity production.

On the hot forming side, hot rolling of thick sections can be considered a rival process to Cold Roll-Forming of thin sections. Traditionally, structural components are mainly made up of hot rolled sections. However, hot rolled sections are in fact becoming less popular than their cold rolled counterparts in many applications due to their less appealing properties except in certain situations like corrosive environment where thicker sections may have a definite advantage.

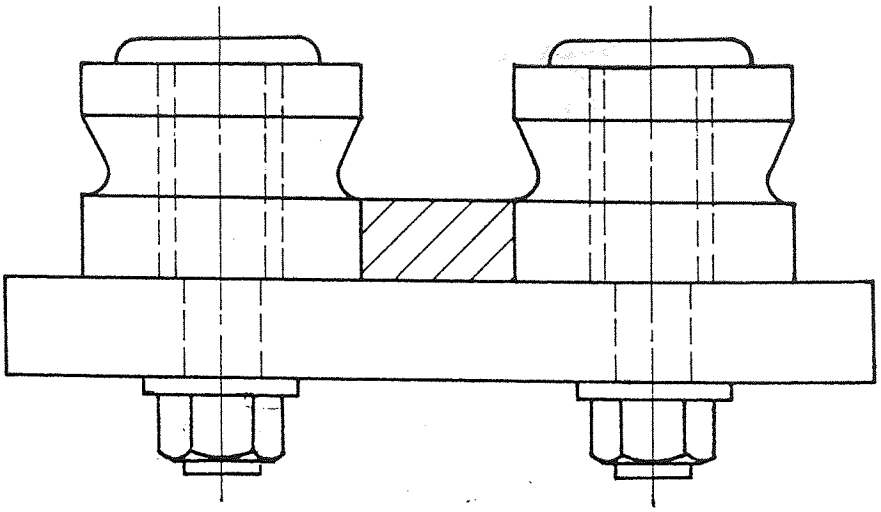


Fig 2.1 AN EXAMPLE OF THE IDLER-ROLLS

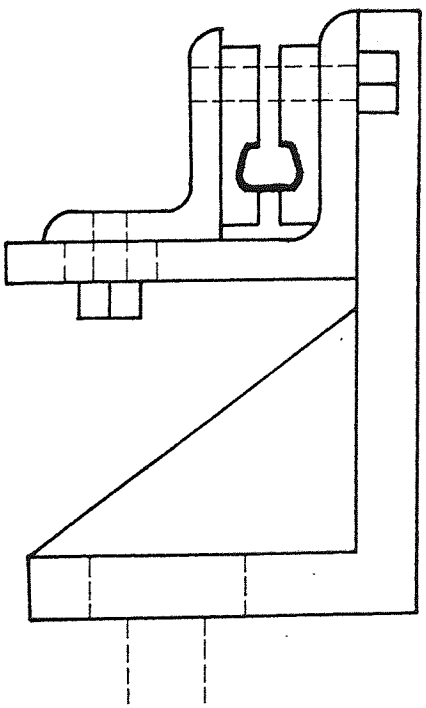


Fig 2.2 AN EXAMPLE OF THE STRAIGHTENING-DEVICE

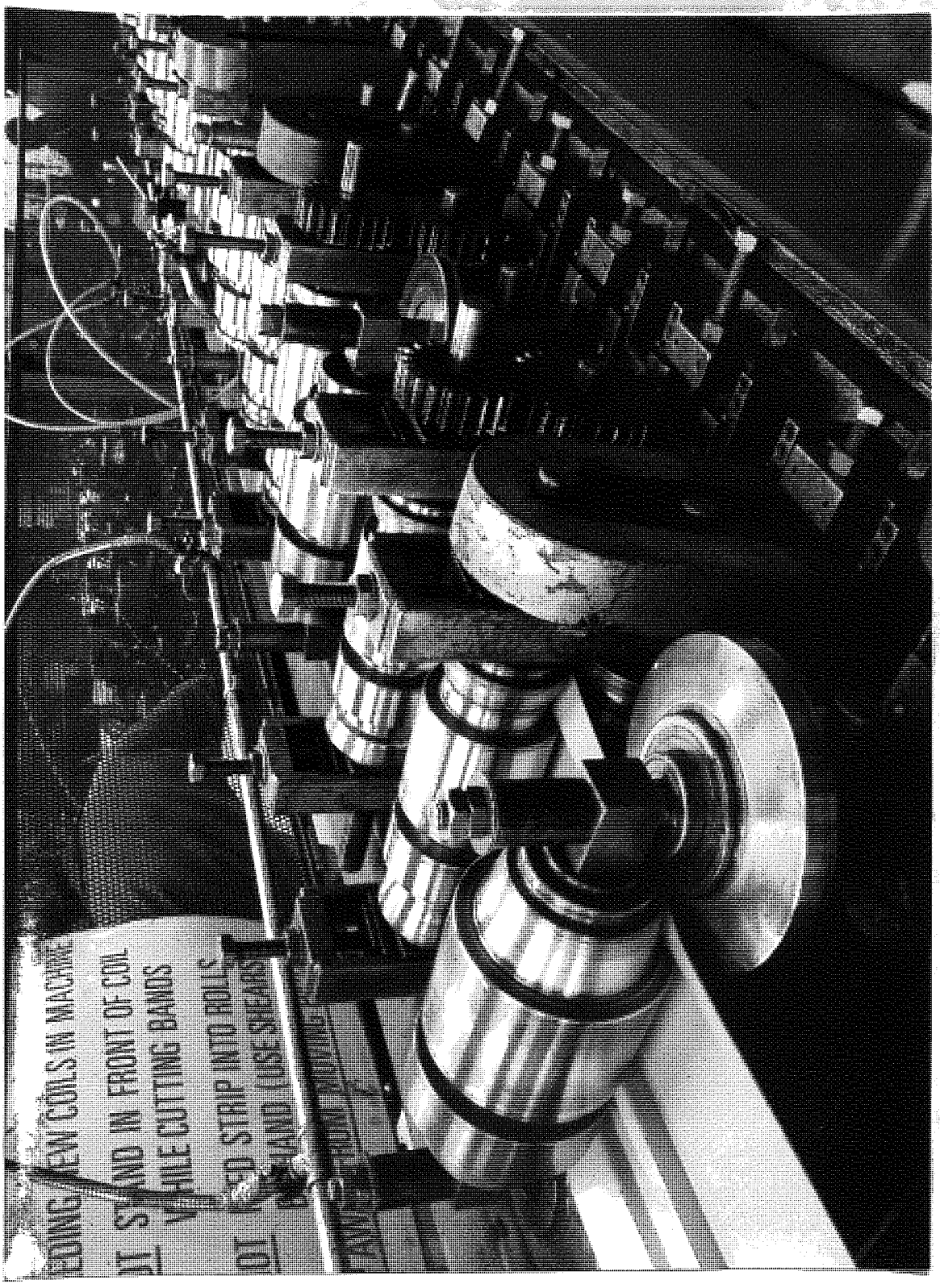


Plate 2.1 COLD ROLL-FORMING

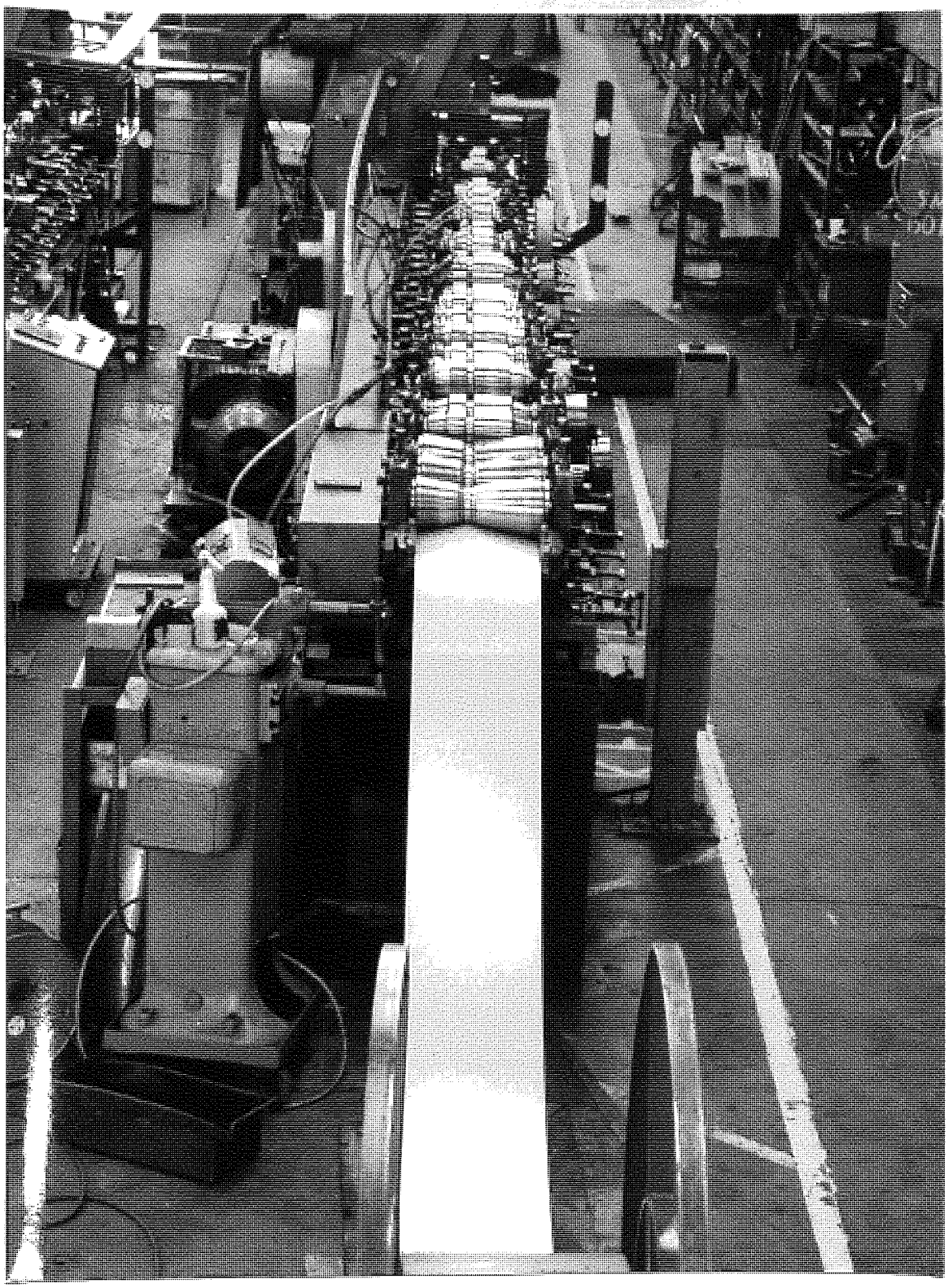


Plate 2.2 A TYPICAL COLD ROLL-FORMING MACHINE

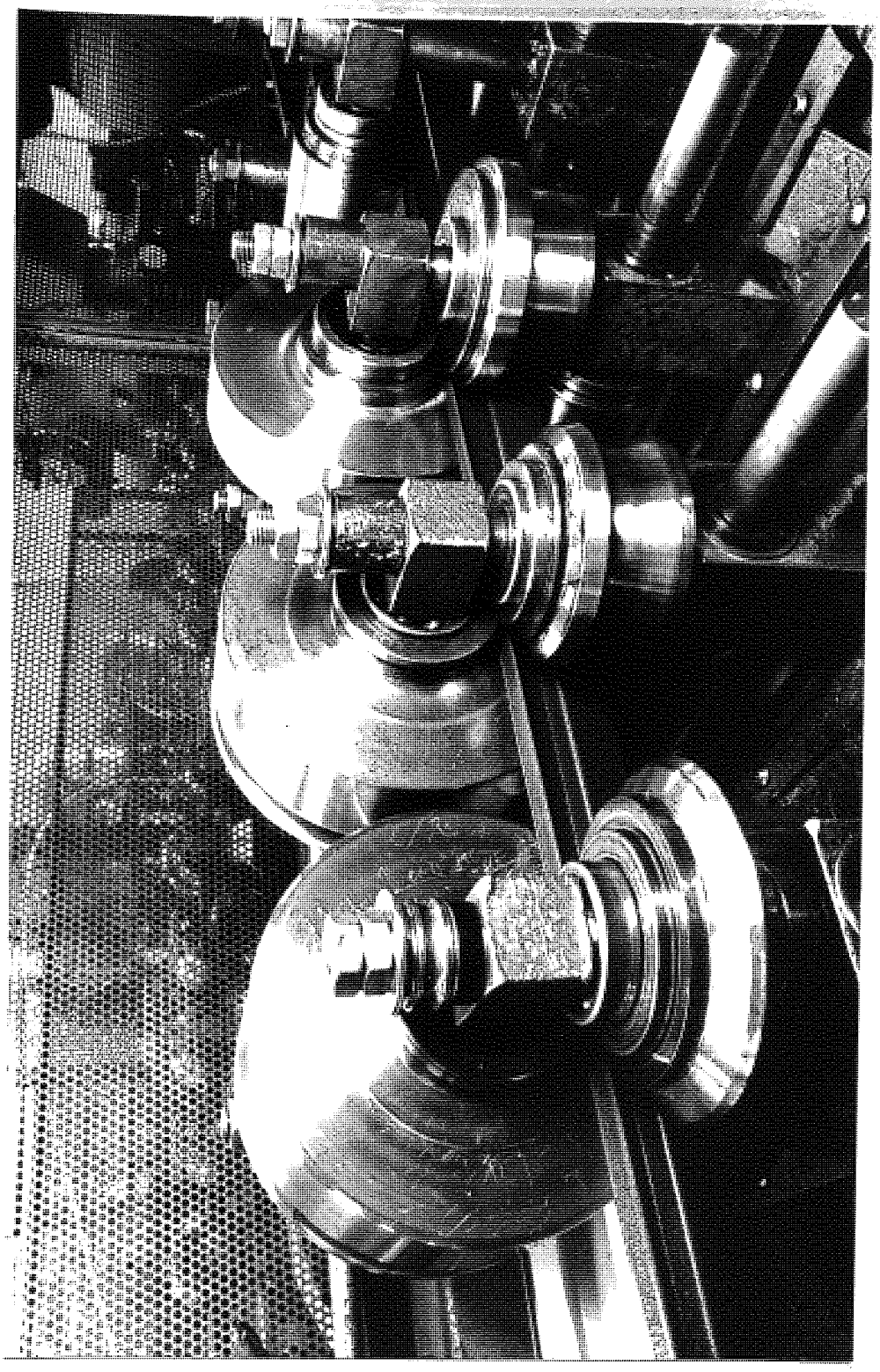


Plate 2.3 COLD ROLL-FORMING STATIONS

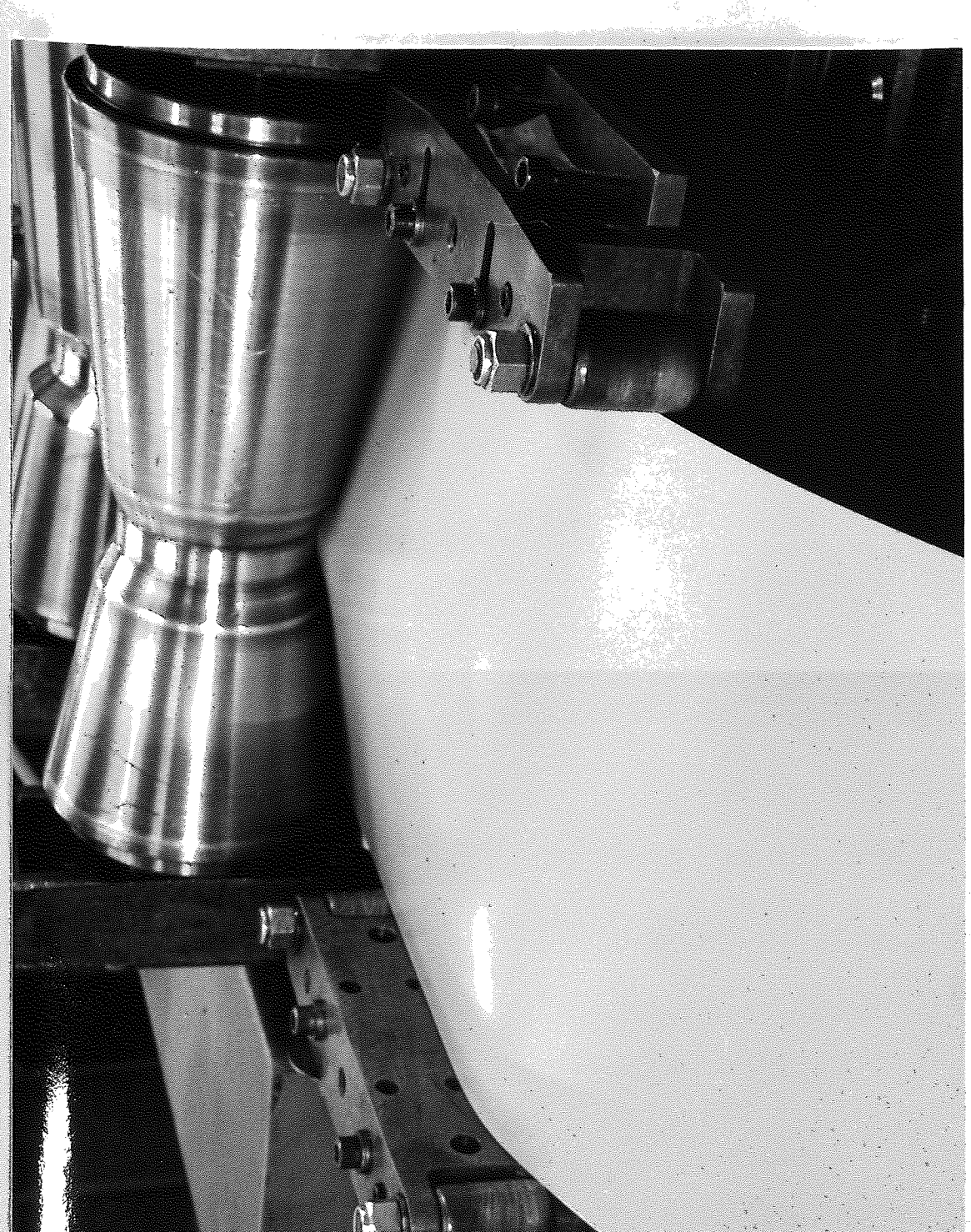


Plate 2.4 A STRIP GUIDE

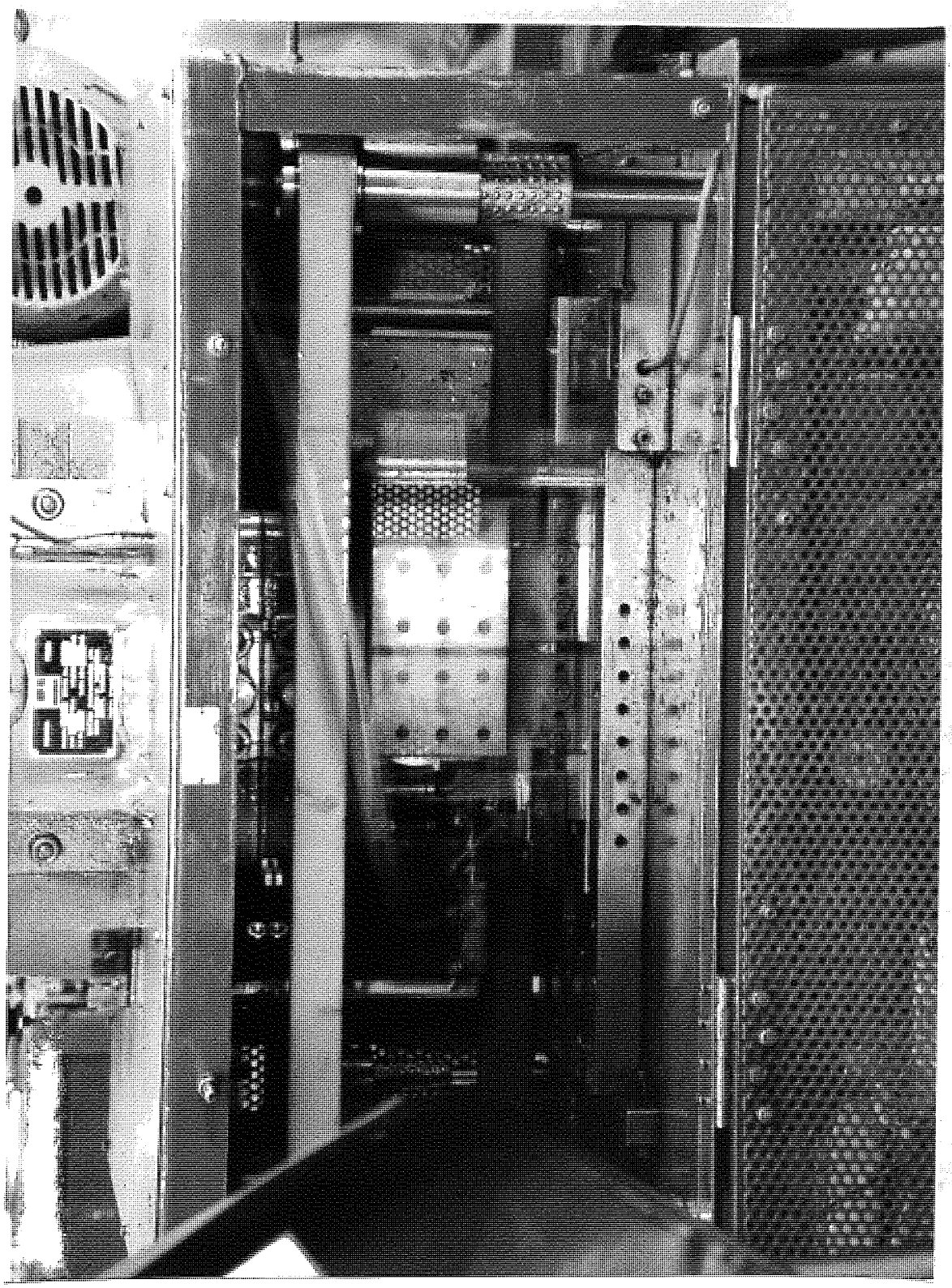


Plate 2.5 THE CUTTING-OFF OPERATION

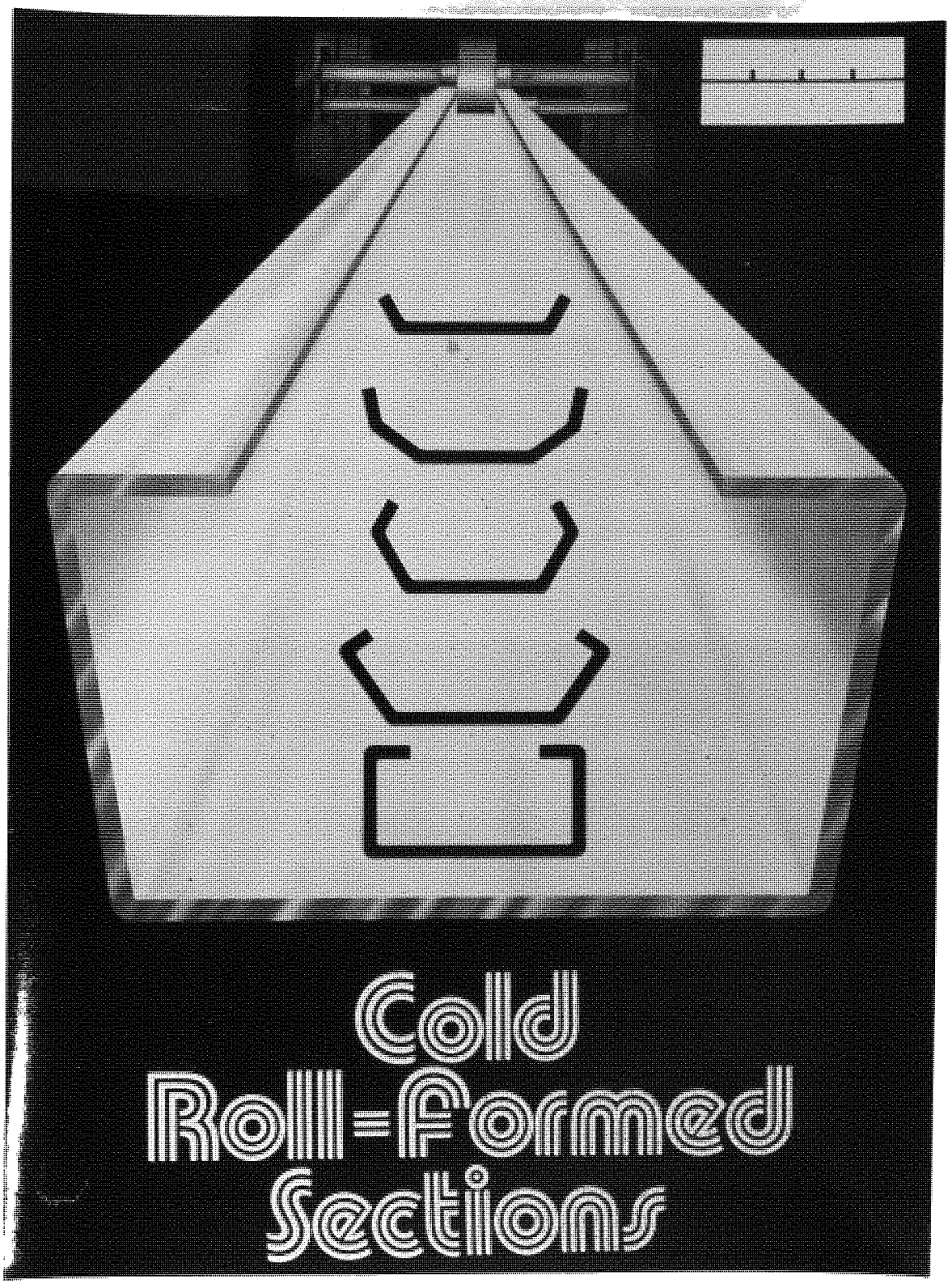


Plate 2.6 THE COLD ROLL-FORMING OPERATION

CHAPTER 3

CONVENTIONAL FORM-ROLL DESIGN AND MANUFACTURE

3.1 Introduction

The design process is often treated as an art rather than a science, even in engineering, which is essentially a discipline of scientific nature. That is understandable because probably very often practice is already urgently needed before any corresponding theory can even be found or established. Empirical experiments with trial and error usually prevail before reliable experiences could be accumulated. Although the design methods may appear unscientific in many cases, the fact that they do stand the test of time is often sufficient to justify their very existence. It is therefore the author's view that conventional design methods, and for that matter manufacturing methods as well, should not be abandoned lightly in favour of new methods unless the latter are equally or more promising.

Designers who in the past were severely restricted in the choice of ways to tackle their jobs can in theory be relieved of such plight when new ways are available. The new ways can often be brought about by new techno-

logical breakthroughs, in this case the use of computers. However, before any such new ways can be established, it is generally necessary to carefully analyse and examine the existing ways, for the sake of thoroughly understanding their nature and for pinpointing the areas of both efficiency and inefficiency. The findings usually could be very helpful in the pursuit of new and better solutions for the old tasks and problems. It was with such an open mind that investigations into existing methods of roll design and manufacture were conducted.

There are four major and distinctive areas constituting the entire roll design activity to be considered, based on current practices of the company. They are:

1. DESIGN OF THE FINISHED SECTION

2. DESIGN OF THE FORMING SEQUENCE USING FLOWER PATTERNS

3. DESIGN AND PRODUCTION OF WIRE TEMPLATES

4. DESIGN OF THE ROLLS

They are described in some detail in the following sections and will be discussed more fully in later chapters where appropriate.

3.2 Design of the Finished Section

The finished section is the product which the whole process of roll design and manufacture aims to accomplish. A typical example of the finished section is shown in Fig 3.1. Until a design of the finished section is available and accepted, it is normally impossible to initiate the roll design process. This is so because the entire roll design work has to be based upon the shape or geometry of the section. Roll designers normally do not produce a section design for the required product themselves, that is in fact strictly the task of section designers who have quite a different set of objectives to achieve. Section design criteria are often application orientated and are based on product requirements, which may be for structural constructions or for decorative purposes. Thus frequently sections are primarily designed for their strength or finish or both, with satisfactory combinations of vital features like shape, size, span, weight, material, cost and so on. Although section design itself is an important field in the thin-walled section manufacturing industry using cold roll-forming, it is however not the primary concern of this research. Details of usual practices employed in section design for various applications can be found in other more

relevant literatures[17 - 20].

When a section design becomes available, either from the customers or within the company, roll designers are then required to scrutinize the design and judge from their own experience whether it could be cold rolled satisfactorily. Routine checks have to be made on each given section design against the capabilities and limitations of the existing process and facilities. Modifications to the section design may be necessary if in its given form it is likely to cause problems. For instance, one of the routine exercises performed by roll designers on every given section is the calculation of the raw material strip width for forming. The strip width has to be checked against the width of the available machines to make sure that it is well within limits as otherwise forming the section would be impossible.

As illustrated in (Plate 3.1), the designed sections can often come in a great variety of shapes and sizes. Nevertheless, they do exhibit some common features which deserve attention. As far as the general shape or geometry of these sections is concerned, apart from being uniform in thickness, it is made up of only two kinds of elements, namely, linear elements and circular elements.

Other kinds of non-linear elements are hardly ever used in practice, probably because they are no better than circular elements in terms of applications and properties or perhaps because they are difficult (and thus costly) to design and manufacture. In view of this fact, only the use of linear and circular elements have been considered in this research. However, should the unlikely need for the use of other kinds of non-linear elements arises, additional considerations would be necessary or alternatively, the new kinds of elements may be approximated using circular elements.

One final acceptance test on a given section design occasionally performed by roll designers is to bend a wire template according to a 10 to 1 scaled up drawing of the section. The philosophy behind it is obviously that if one can achieve the required shape using wire templates satisfactorily, it should be all right to cold roll the same shape from the strip.

3.3 Design of the Forming Sequence using Flower Pattern

The forming sequence is a collection of the intermediate section shapes which the strip progressively assumes as it is passed through the rolls at each forming station. Given an acceptable section design,

with sufficient experience roll designers normally can produce a reasonably satisfactory forming sequence for most of the ordinary shapes. The basic things to be decided at this stage are:

1. The number of roll stations required for forming and ironing.

2. The sequence and degree of bending at each station for the circular elements.

3. Whether any inclination of the strip is necessary for better forming.

The usual tool used by roll designers to represent forming sequence is the drawing of flower pattern or flower plan. The flower pattern is a pattern showing progressive development of the section shapes in successive stages of forming, from start to finish. A typical flower pattern is shown in Fig 3.2. With the flower pattern, the distribution of bending of each element at each stage can be easily traced and can be repeatedly modified if necessary, until a satisfactory forming pattern is obtained. The final form of the flower pattern can then be used as the equivalent of the required forming sequence for the section.

The following factors are usually considered in the design of forming sequences:

1. SECTION SPECIFICATIONS

(a) Shape Complexity

Complex shapes have more parts to be bent and therefore require a greater number of roll stations.

(b) Material

Materials of high yield strength are less formable and therefore also require a greater number of roll stations. Usually, to form such materials, greater driving power is also required at the rolls and that in turn limits the amount of bending possible at each roll station.

(c) Tolerance

Tighter tolerance usually requires more precise forming and also additional finishing stations.

(d) Size

Larger sections normally require the use of larger rolls and as a result the cost of roll material may become significant. For certain section shapes, it is sometimes possible to incline the strip instead of feeding the strip horizontally so that smaller rolls could be used, thereby reducing the cost of roll material. Alternatively, composite split rolls of different materials may be used so that areas of lower wear contain cheaper material.

2. PROCESS REQUIREMENTS

(a) Angle of entry and Strip flow

Normally if the strip enters the rolls in a certain direction, the tendency is that it will continue in the same direction when it emerges at the other side of the rolls. Upsetting the natural tendency of strip flow will normally result in disastrous forming consequences, the strip may wrinkle, buckle or even fail to be formed. The amount of progressive bending

of each element at each forming stage needs to be carefully distributed so that a smooth flow of the strip can be maintained. Although it may not always be possible to maintain a very smooth flow of the strip, especially with complex shapes, it is still desirable to avoid as much interruption to the flow as possible. Suitable adjustments of the pass heights of the strip from roll station to roll station can usually improve the strip flow as well.

(b) Cumulative angle of bending with respect to Horizontal Axis

In general, angles of less than 45 degrees can be formed readily with top and bottom rolls alone, but for angles exceeding 45 degrees, forming is more satisfactory using side-rolls. Side-rolls also have less tendency to interrupt the side way flow of the strip during entry to the rolls.

(c) Radii of Bending

The inside radii of bending have a

significant effect upon forming. Sharper radii can overcome spring back and slipping of the material more effectively but wear out faster. When bending very short legs, typically of length less than five times the thickness, the use of sharp radii is a necessity. There are a few different ways of controlling the radii of bending. Some designers may prefer constant radii from stage to stage while others may prefer varying radii, and still others may prefer an appropriate combination of both, whichever suits them best.

(d) Cutting-off Requirement

Forming pattern using upward bending of a horizontal strip is normally preferred. However, the necessity to cut-off the finished section after forming at a particular orientation may require the use of downward bending.

3. PRODUCTION REQUIREMENTS

(a) Production Quantity

In general, for large production quantity, the tendency is to use more roll stations than the minimum number required so as to cope with wear and tear, whereas for small production quantity, usually only the minimum number of roll stations is used, to cut down the cost of rolls.

(b) Production Rate

If high production rate is required, forming speed can be increased by doing less bending at each stage, with more stations.

3.4 Design and manufacture of the Wire Templates

With the required forming sequence, which is in the form of flower pattern, a set of wire templates (Plate 3.2) can then be designed and manufactured accordingly. The wire templates are created as intermediate simulated sections of the strip at each successive forming stage and are used subsequently as the master contours or shapes in roll machining. Hence

they in fact form an integral part of roll design and manufacture.

The technique used to design and produce the wire templates is simple, but practical. Based on the flower pattern, the shape of each element bent to the required angle at a particular forming stage is worked out and all of the element shapes are then grouped together to form the required section shape at that stage. At that point, usually the element dimensions and shapes are adjusted in a certain manner so that the rolls machined according to the templates can control the material flow and distribution more effectively during forming. The simulated section shapes are usually drawn ten times as large as the actual size. With the help of a shadow-graph projector, or perhaps some other similar equipment, straight wires are bent to the exact simulated shapes according to the 10 - 1 template drawings (Plate 3.3). Bending using drawings of size larger than the projector screen is performed in several parts, with the help of markings. The entire process is manually operated and some considerable skill in wire bending is necessary.

3.5 Design of the Rolls

Unlike traditional approach, which uses very detailed

roll drawings, only simplified roll drawings are produced and used. The simplified drawings carry information such as the overall roll dimensions, roll materials, general tolerances for machining, pinch-difference surface allowances, the use of side-rolls or spigots and approximate shape of the rolls. Each roll drawing corresponds to each forming stage and is used to complement the wire template during machining of the rolls. Roll diameters are adjusted at the time of drawing to give a smooth variation of strip pass heights from station to station for better forming. Other means of guiding the strip flow are also incorporated at the same time if necessary.

3.6 Manufacture of the Rolls

The rolls are normally machined manually on conventional lathes (Plate 3.4). The accuracy of the roll contours so produced relies heavily upon the skill of the machinist whose only guides are the wire templates and the simplified roll drawings. Much discretion has to be exercised by the machinist during machining and mistakes are normally irrecoverable. The success of existing practice therefore depends upon consistent performance from machinists with the right skill.

3.7 Discussions

From the above analysis it was found that the most inefficient aspects are those which require persistent human effort and those which are prone to errors. The mundane work like calculations, drafting and machining can to an appreciable extent be handled by some computerised methods. Design decisions to some extent can also be made automatic but more detailed considerations are necessary. The whole computerisation approach will be discussed more fully in subsequent chapters.

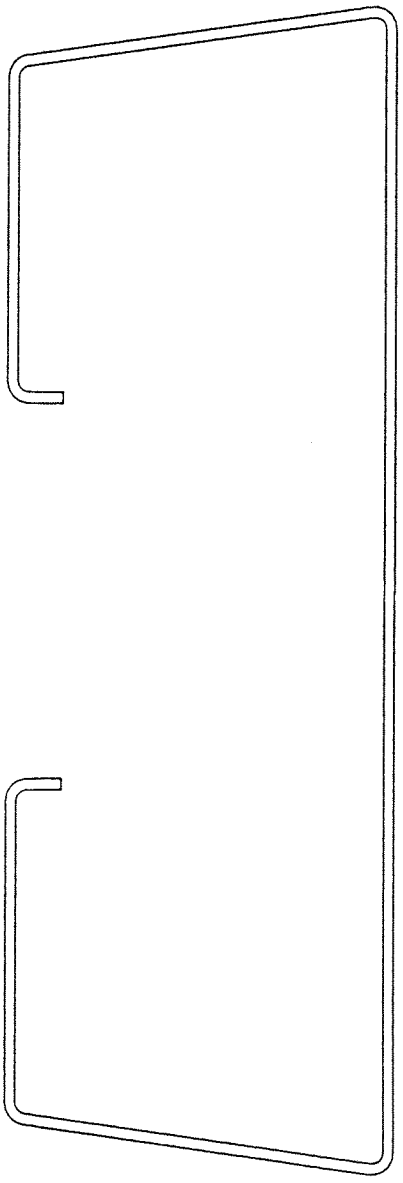


Fig 3.1 A TYPICAL FINISHED SECTION

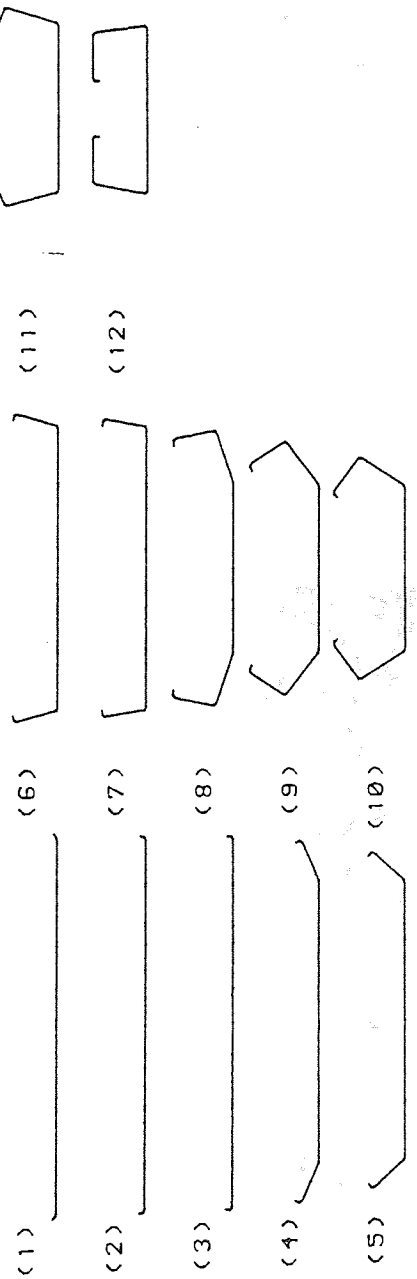


Fig 3.2 A TYPICAL FLOWER PATTERN

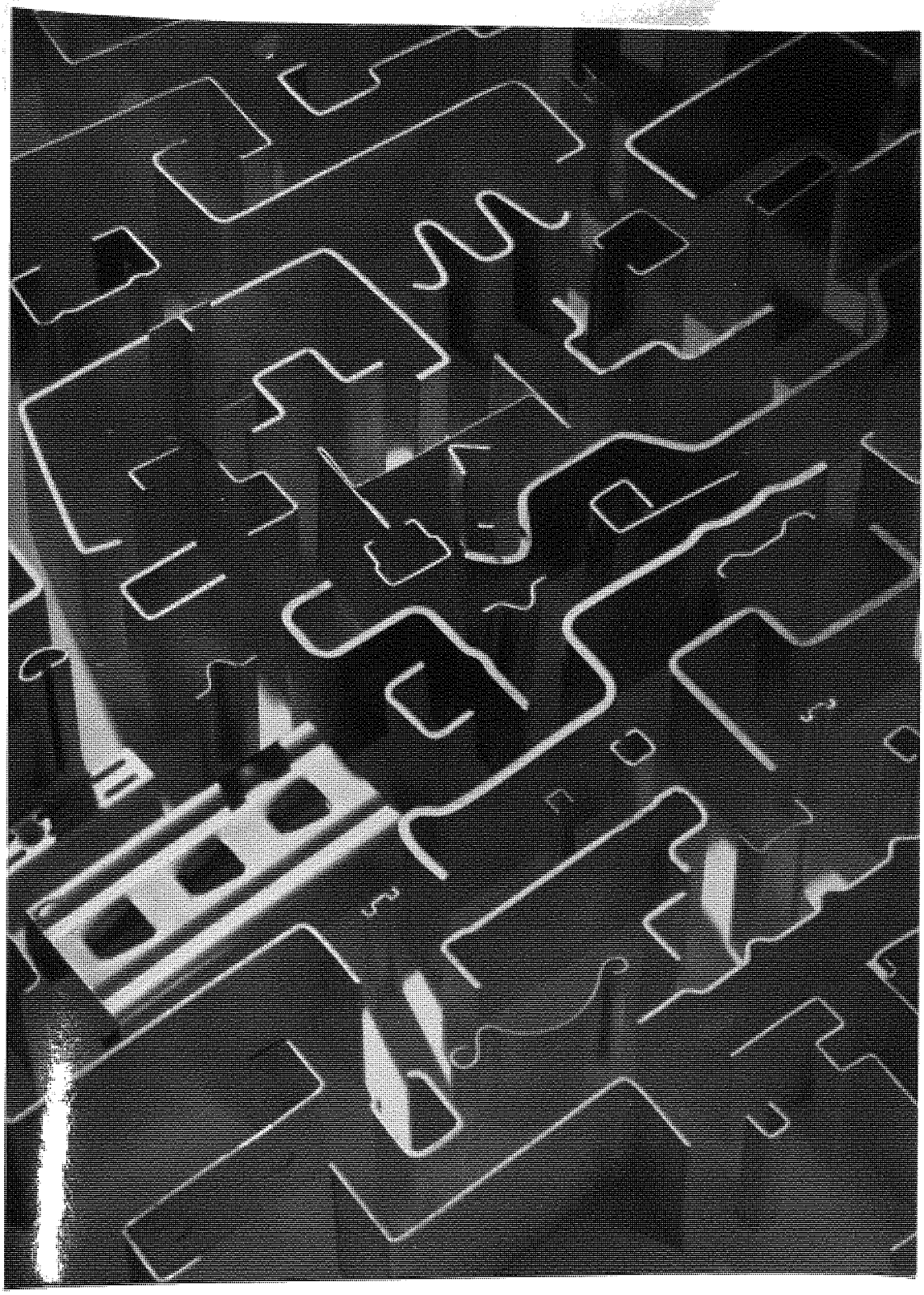


Plate 3.1 EXAMPLES OF THE FINISHED SECTION

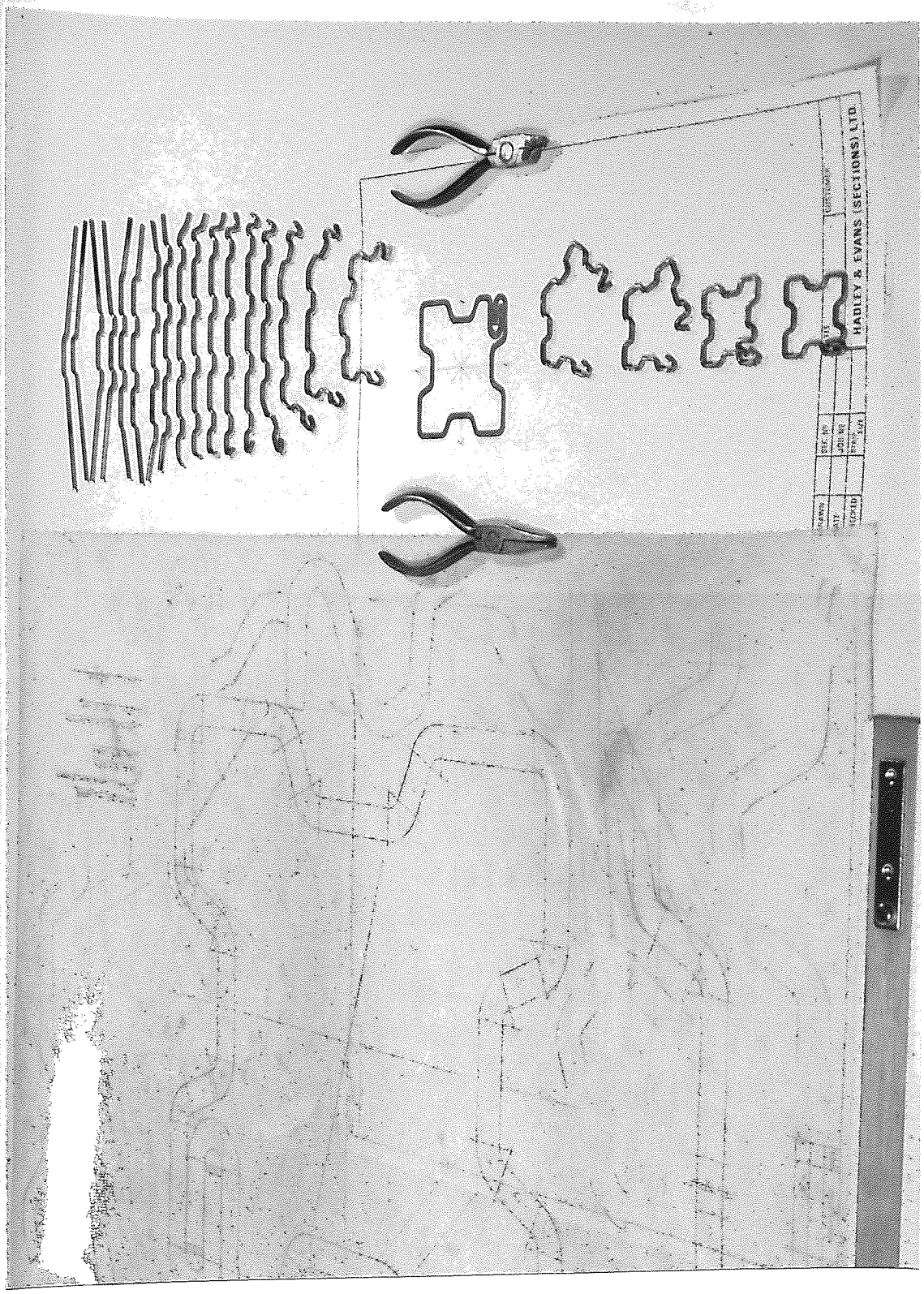


Plate 3.2 THE WIRE-TEMPLATES AND A 10 TO 1 TEMPLATE DRAWING

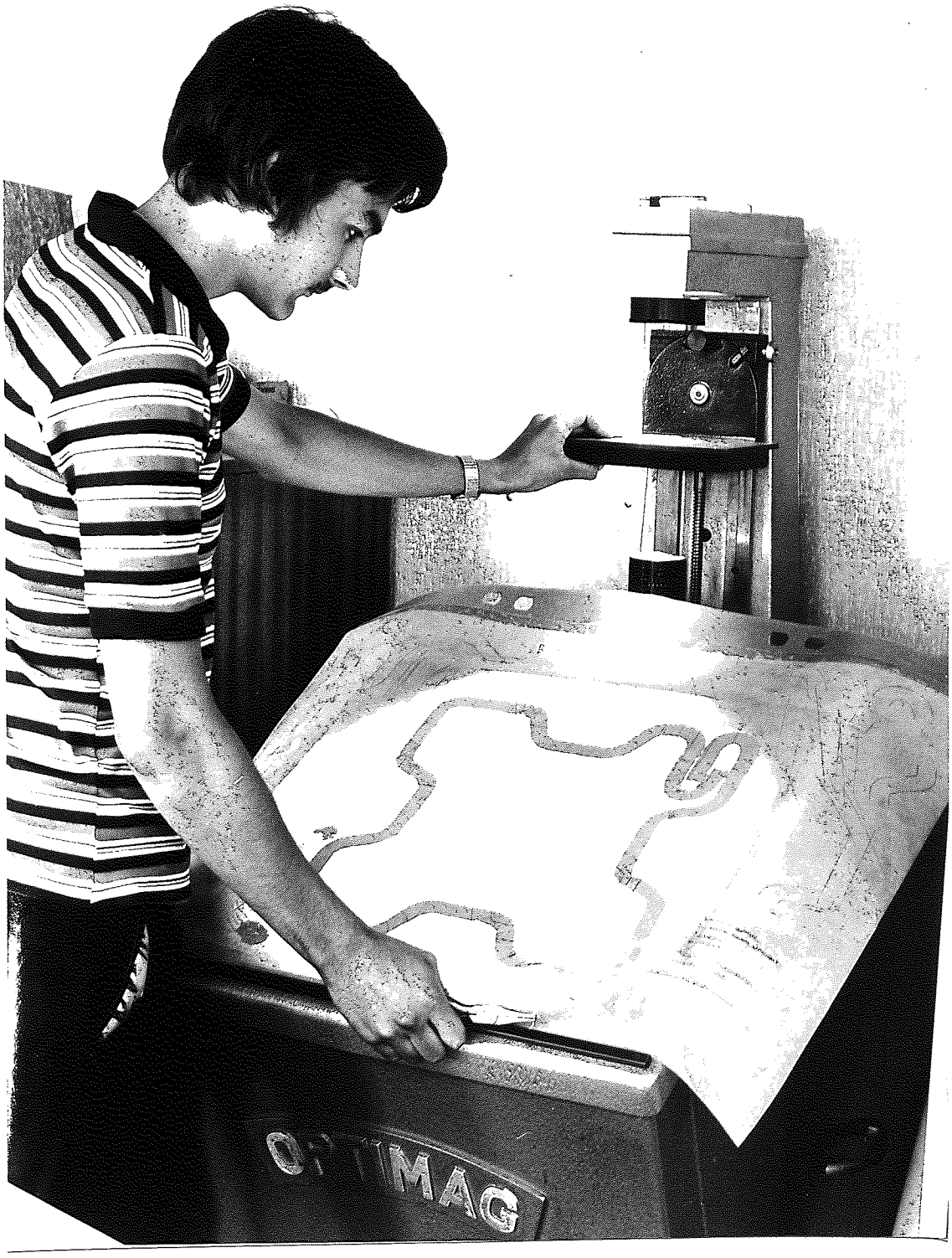


Plate 3.3 PRODUCTION OF THE WIRE-TEMPLATES ON A SHADOWGRAPH PROJECTOR

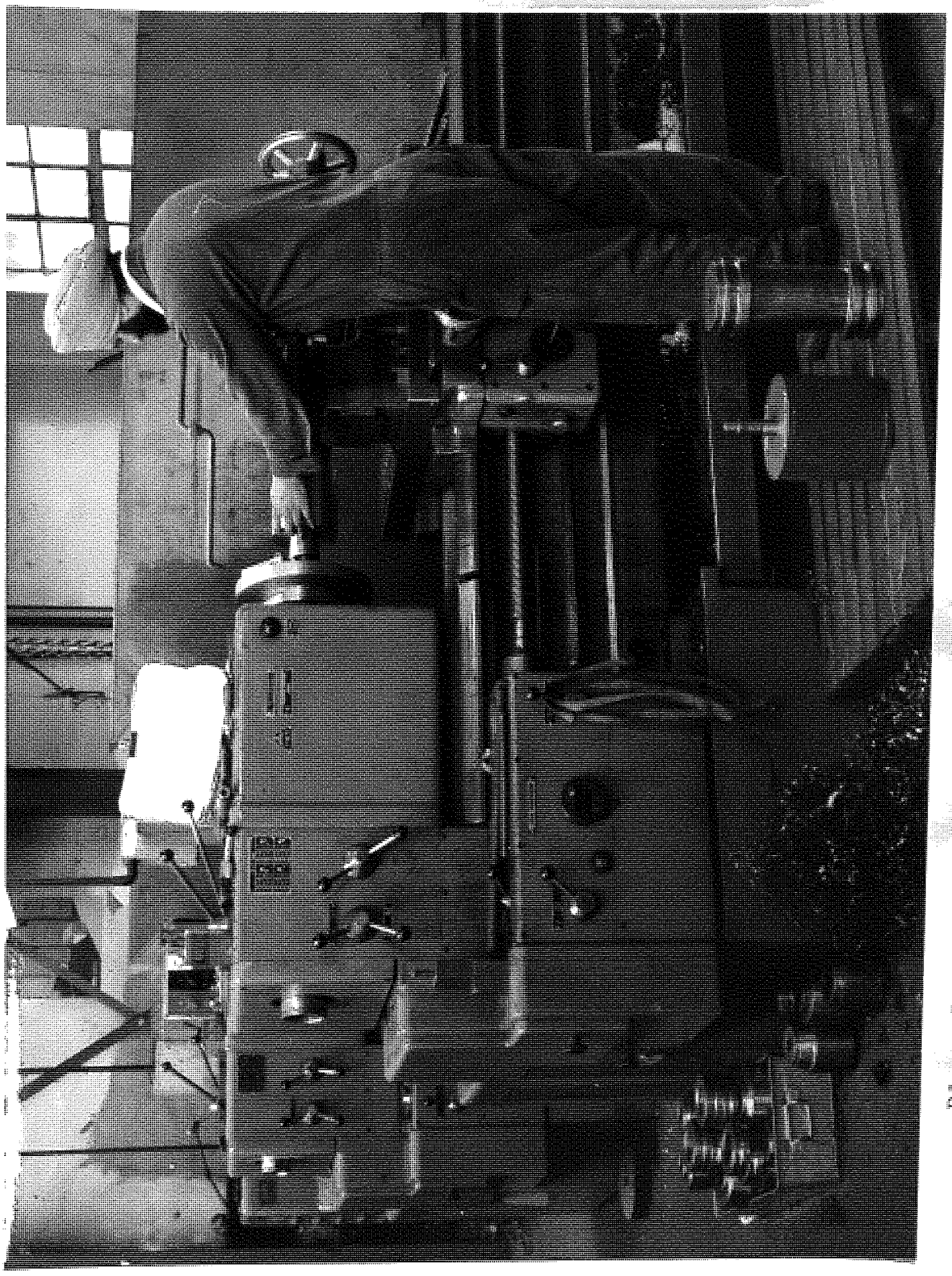


Plate 3.4 FORM-ROLL MACHINING ON A CONVENTIONAL LATHE

CHAPTER 4

PLANNING FOR THE COMPUTERISATION OF FORM-ROLL DESIGN AND MANUFACTURE

4.1 The General Advantages of Computerisation

The word computerisation is used here to denote the application of a digital computer to carry out either part or whole of a function or process, such as design and manufacture. The basic structure of a digital computer is shown in Fig 4.1. Computers falling into this category are numerous, they include very small microcomputers using microprocessors, mini-computers, main-frame computers and very large main-frame computers. Normally the larger the computer, the more powerful is its capability and the more complex is its range of facilities.

The basic functions of a digital computer are to input and store specific data and instructions, to perform logical and arithemetic functions with the stored data according to the stored instructions and to output the resultant data. The stored data and instructions are collectively termed software whereas the physical components which carry out the computer functions in automatic coordination are collectively termed hardware.

Software structure can be altered freely while hardware structure is often of a permanently fixed design using standard building blocks. For a particular application, software elements may be arranged and built up in a particular manner so that the computer can be conditioned to perform the specified task.

The generally accepted advantages of using digital computers are as follows:

1. SPEED

   The speed of calculation by computers is very much faster than the human speed.

2. ACCURACY

   Computers are much less error-prone than human beings and they tend to produce more consistent results.

3. MEMORY

   Computers can be designed to memorise as much information as required and their memory is reliable and does not fade with time.

4. FAITHFULNESS

Computers carry out instructions without questions and require no motivation.

5. FATIGUE FREE

Computers do not get tired easily and need not rest like human beings but imperfection in present hardware technology do contribute to computer break-downs which can usually be prevented by proper hardware maintenance and update.

6. AUTOMATIC

Once in operation, very little human interference is required. Operators are needed only for house-keeping tasks.

7. CAPABLE

Jobs too complicated, too extensive in scale or taking too long a time for human execution may be handled easily by computers.

There are however some pitfalls which have to be guarded against:

1. The costs of purchasing, operating and maintaining computer equipment may turn out higher than the costs of employing comparable skilled personnels.

2. Selection of suitable computer equipment for the required job may be difficult. Consequence of mistakes in selection could be expensive.

3. The desired computer equipment, both hardware and software, may not be available. Specially developed one-offs are usually costly and take time to complete.

4. In the event of computer failure during service, consequences can be disastrous or even fatal. That however can usually be prevented by taking adequate precautions.

The pitfalls listed above can normally be avoided with sufficient care. All the points discussed so far about the use of digital computers are generally true for any kind of application. The fact is that computers are

becoming more and more indispensible to man, just like any other technology man had developed and learned to use in the past.

Both Computer-Aided Design (CAD) and Computer-Aided Manufacture (CAM) industries are currently advancing very rapidly in both their scope of influence and their technological expertise. Competition among rival enterprise is likely to escalate with time in the exploitation of CAD and CAM in order to outdo one another. That could well lead to a survival battle in the end. Those far-sighted organisations usually act early to be ahead. Besides, early computerisation tends to cost less and bring about benefits early.

4.2 The Role of Computers in Design and Manufacture

The general idea of using computers is to help man perform tasks that are manually inefficient, undesirable or even impossible. Certain qualities of computers as discussed earlier are difficult to match with human effort. Computers can therefore in general play a very useful role indeed, especially in areas where human handling had failed to work satisfactorily in the past.

In both design and manufacture, time is an important factor because it affects things like expenses, delivery period and availability of tied-up resources and facilities for other usage. If a substantial part of a manually operated process can be successfully computerised, time normally required with manual effort can then be reduced considerably. There are many such processes where computers can come to aid. For instance, complicated data calculations when handled manually are liable to frequent mistakes, which usually take time to trace and correct and which can be costly if they are not discovered early enough. Reducing chances of such mistakes from occurring by using computers can in turn reduce the chance of unnecessary wastage of time, effort and other resources. Another example is performing tasks that are unpleasant, such as hazardous tasks, boring tasks, tiring tasks, continuous tasks and so on. They normally have adverse effects upon human performance but probably do not affect computers at all. Jobs which are too complicated, too big in scale or too fast for human handling very often must also rely on the use of computers. However, present level of computer technology still do not offer the kind of human capabilities like intelligence, pattern recognition and so on, jobs

demanding these capabilities are still somewhat formidable to computers.

For both design and manufacture, there are already a wide range of computerised products commercially available for acquisition, whether in the form of hardware, software or integrated systems. Typical examples of computerised design aids are automatic drafting machines, integrated graphic systems, design software packages for specific applications and so on. For manufacturing purposes, there are highly automatic machines and systems such as Computer Numerical Control (CNC) machines, Direct Numerical Control (DNC) systems, Flexible Manufacturing Systems (FMS), robots and so on. There are also computer aids for processing manufacturing information, like the Automatic Programmed Tools (APT) language and system for Numerical Control (NC) part-program tape preparation and other kinds of language and systems for the same purpose, production control and stock control systems, technological database systems and so on. More of such design and manufacturing tools with increasing sophistication are likely to be available in future.

4.3 Alternative Ways of Accomplishing Computerisation

Although in the past computerisation of an existing process required complete evaluation of both hardware and software, in these days computer hardware tends to be more standardised, and so is part of the computer software system. Such standardised computer systems may be combined with specially written user software or application software for specific applications. As far as design and manufacture are concerned, there is already a wide range of computerised systems and aids available as described earlier. Most of these systems, however, are either too generalised or too specialised and it is rarely possible to obtain a system which completely satisfies the requirements of a highly specialised job, like form-roll design and manufacture. The required system can of course be in the form of spceially designed hardware and software or standardised hardware and partially standardised software with specially designed application software. The latter normally turns out less expensive and more easily obtainable. The application software itself can be purchased or hired if it is commercially available, otherwise it has to be designed and developed to specifications. Both purchase and hiring could run the

risks of obtaining software which is not suitable for the purpose or which is too costly to implement. Software may normally be obtained from sources like the computer manufacturers, government agencies, research and development bodies, specialised software houses, commercial computer service bureaux, engineering consultancies, even academic and private establishments and so on[22]. As far as the author was aware, the only organisation which had developed any software in the field of roll design and manufacture was the Machine Tool Industry Research Association (MTIRA)[23], however, the software developed there was of limited capabilities and did not meet the specific requirements of the company. In the absence of suitable software for acquisition, the only suitable solution then was to computerise the roll design and manufacturing process by software design and development, so as to produce a special application software package which could be implemented on a computer system of the company's choice. Systems with graphic capabilities suitable for this purpose currently available are the minicomputer based integrated graphic systems which are better known as the TURNKEY systems (or sometimes the CAD systems)[24]. The much smaller DESKTOP systems[25] can also be used

provided their range of facilities and memory capacity are adequate for the required software implementation. The main advantages associated with computerisation by software design and development in general are:

1. The software can be tailored to exact and specific needs.

2. The software can be readily extended to cater for new needs or to handle other related tasks.

3. The software can be gradually incorporated into existing process without serious disruptions as development proceeds.

4. Users can be continuously and intimately involved throughout the development for more satisfactory design of the software with regards to its usability.

The main drawback of the software design and development approach is that it takes time to produce the required software and hence requires careful planning.

4.4 General Considerations for Computerisation by Software Development

Software development, like any other development work, requires systematic planning, especially when the software involved is of some appreciable scale and complexity. A set of software objectives and computerisation requirements based upon the existing needs against the constraints of the available resources, facilities and knowhow has to be evaluated and established to guide the whole development process. All significant factors of influence on all relevant aspects of computerisation must also be taken into account for efficient development. The factors which had been considered are outlined in the subsequent sections and will be elaborated later on where appropriate.

4.4.1 Scope of Computerisation

There are two broad aspects to be considered regarding the scope of computerisation. One is the horizontal aspect, namely the extensiveness of computerisation, or the total number of different functions to be covered during development. The other is the vertical aspect, namely the depth of computerisation,

or the extent to which computers are to replace human effort in the process to be computerised. Both aspects have to be limited somehow in practice because they both have decisive influence over the consumption of resources and the use of facilities, and also because they both are limited by the existing level of knowledge and technology. For the purpose of this research, only those functions which were relevant to roll design and manufacture and which could be computerised were computerised, subject to selection for maximizing benefits and feasibility.

4.4.2 Assessment of Needs

For successful computerisation a clear understanding of existing practices in the company and the company's needs was necessary. To accomplish that frequent and comprehensive discussions had been held with all relevant personnels of the company in preparing a suitable and beneficial programme for computerisation throughout this research. Priority was given to the satisfaction of more pressing primary needs, with the secondary needs considered only when the primary needs had been satisfied.

4.4.3 Computerisation Methodology

This involved the adoption of suitable and methodical techniques in deriving the computerised equivalent of the existing process and in designing and developing the corresponding software as a substitute. Existing roll design and manufacturing activity in the company had to be analysed and assessed so that promising areas for computerisation could be established. The software tools were also selected for their relevance in the development, more will be discussed in the subsequent chapter.

4.4.4 Development Sequence

Strategic measures were taken in adopting a development sequence which suited both the short-term and long-term requirements of the company for computerisation and which contributed significantly to efficient development. That was necessary to avoid upsetting existing operations in the company and to reduce the chance of resources wastage and delay in completion, more of these will also be discussed later on.

4.4.5 Available Resources

The most vital physical resources involved in software design and development in general are time and labour. In practice both are finite and therefore should be utilised as efficiently as possible. In this research they were mainly channeled to the analysis of existing roll design and manufacturing activity and the design and development of the required software.

4.4.6 Available Facilities

Facilities were required mainly for software development purposes in this research and they consisted of a computer system complete with the relevant compiler, editor, system software, supporting software and relevant peripherals. As the production of design drawings was necessary, graphic software and hardware had to be used. At Aston, graphic facilities in the form of GINO-F software library and graph plotters were available with the ICL 1904S computer system. Other relevant facilities were not available then.

4.5 General Planning for the Computerisation of Form-Roll Design and Manufacture

After thorough investigation into the existing roll design and manufacturing activity and detailed discussions with all the relevant personnels concerned in this research, the following areas were selected for computerisation:

1. Production of the finished section drawings.

2. Design and production of the forming sequence using flower pattern drawings.

3. Design and production of the 10 to 1 template drawings for existing manufacturing use.

4. Design and production of the roll drawings and roll-contour data for future manufacturing use.

5. Roll contour data processing and roll manufacturing.

The primary needs and secondary needs of each area of the activity were examined for their urgency so that development could be planned accordingly. The following stages of development were planned:

FIRST STAGE

Design and development of software for automatic drafting of the finished section, the flower patterns and the 10 to 1 templates, with the required design definitions and control features incorporated. Conventional methods of roll design and manufacture will continue to operate, with the design of the forming sequence and 10 to 1 templates partially handled by the developed software.

SECOND STAGE

Design and development of roll design software, incorporating as many routine roll design techniques as possible, for NC manufacture. This when successfully developed and implemented will ultimately replace the use of 10 to 1 templates for roll manufacturing purposes. Roll contour data generated by software will then be processed and used in NC tape preparation. For interim manufacturing purposes, NC lathe and some computer-assisted part programming system can be acquired to partially automate the roll manufacturing process.

THIRD STAGE

Design and development of software for non-routine roll contour editing and automatic preparation of NC tapes for manufacturing. Roll design process can then be linked to the roll manufacturing process that uses NC lathe(s) and implemented as an integrated CAD/CAM system.

The above development plan was prepared for both the short-term and long-term needs of the company and the up to date results of its implementation were encouraging. So far the first and second stage software development have been completed and the company is currently acquiring the relevant equipment for the interim implementation. Both the design and manufacturing interests of the company were actually enhanced progressively throughout the development.

Although the use of wire-templates made from the 10 to 1 template drawings originally intended for roll manufacturing purposes is likely to be replaced by NC methods in future, nevertheless the wire-templates still serve as a useful form of quality control tooling. During the transitional period, manufacturing with

conventional lathes and with NC lathe will co-exist for some time until the NC lathe is in a position to handle all the roll manufacturing jobs as competently. When the integrated CAD/CAM system is completed and properly utilised, roll design and manufacturing productivity should improve significantly.

The following sequence of software design and development was planned accordingly as the entire computerisation programme:

1. Design and development of the finished section software.

2. Design and development of the flower pattern software.

3. Design and development of the 10 to 1 template software.

4. Design and development of the roll design software.

5. Design and development of the roll contour editor.

6. Design and development of software for automatic NC

tape preparation based on the roll contour data generated.

In this research, the first four units of the above listed software had been completed, corresponding to the second stage of development planned as described earlier. They constituted the major part of the entire roll design process to be computerised, the remaining units of software to be designed and developed largely concerns computerised roll manufacturing.

4.6 Systematic Analysis and Assessment of Existing Roll Design Activity for Computerisation

In general computers are suitable only for handling activity which is routine in nature. A routine activity or process may be regarded as one which involves making consistent decisions about carrying out a selection of the predetermined and precisely defined actions under given circumstances according to a fixed and also precisely defined set of rules. Random decisions and random actions are something computers cannot do by themselves when confronted with totally unexpected situations. For analysis purposes a routine process may be divided into three distinctive sequential parts:

1. Identifying the situation or the particular set of conditions to be dealt with.

2. Making decisions according to the predefined set of decision rules.

3. Carrying out the selected course of predefined action.

For effective computerisation all three parts must be precisely defined. Means of identifying each condition in each situation must be established and made available to the computers. Depending upon the number of conditions associated with each situation, the number of decision rules varies. Although in theory it is possible to cover all possible situations or all combinations of the associated conditions, the number of situations may be so large that it becomes infeasible to do so. Hence in practice only a limited set of situations is selected for handling using an adequate set of decision rules while the rest are rejected as impossible to handle.

Subjective human factor normally has a very important

influence over the definability of each part of a routine process. Traditionally, most designers tend to work from experience which was usually accumulated through long period of painstaking effort by trial and error. Such experience might or might not have been based upon scientific principles and all too often it could have been just a matter of personal liking or habits. It is often possible that a designer may know what he is doing without necessarily knowing exactly why. As a result the decision rules associated with the process become indefinable. Another problem is that different designers often have their own different ideas and hence work according to their own individual set of decision rules and probably different courses of action as well, for the same task. Since computers cannot make ambiguous decisions or take ambiguous courses of action, only one set of decisions rules and the associated set of courses of action may be selected among them for use on the computers. Usually the best set is selected if agreement among the designers can be reached. Failing that, it may still be possible to incorporate the equally valid sets as options in the software which the designers can select for themselves. Failing that as well, the process in question will have to remain manual.

In most cases, computerisation can normally be accomplished by direct translation of the existing process into the computer instruction steps especially the more straightforward but tedious calculations. If only part of the process is definable, then only the defined part may be computerised, the undefined part has to be left to the designers' discretion. Occasionally it may be possible to create a new and completely different process to replace the existing one provided the new process can be shown to be reliable and equally acceptable. Designers often have the tendency to go for the approximate but easy methods rather than very precise but much more complicated methods. With the use of computers, methods regarded as too complicated to perform manually and as a result had been disused in the past may become quite simple to apply. New and better methods can usually be established based on sound scientific principles instead of trial and error.

Apart from the subjective influence, certain human qualities and capabilities are hitherto considered irreplacable by computers due to limitations of the current level of computer technology. One good example is complex pattern recognition for extracting significant

information out of an apparent mess. Although such processes are routine and simple for human handling, in most cases they are quite formidable to computers. Therefore most likely such processes will have to remain manual too.

4.7 Definition of Computerisation needs in terms of Software System Characteristics

Software process or computing process in general, when in operation, may be viewed as a process that maps a desired set of output data values to a specified set of input data values in a defined manner as shown in Fig 4.2. This implies that software process, like other processes such as the production process for instance (Fig 4.3), is basically an input-output system, it can therefore be analysed and synthesized using systems concepts. The products in this case are some useful output data or information. The physical form of the data can be anything ranging from signals, numbers, texts, drawings to almost any intelligible representation. It is through these media that communications are possible and that specific needs are satisfied by the use of computers. Therefore in general the output data set can be defined according to the needs to be satisfied and then be treated

as a faithful representation of the objectives of computerisation. The software system functions and the input data set can then be formulated accordingly in order to generate the required output data set. In this way the needs can be precisely defined in terms of the achievable software system output characteristics and the software system in turn can be precisely defined as well.

The needs established in this research and defined in the described manner are listed in the following summaries:

4.7.1 Software Functions and Output Requirements

1. THE FINISHED SECTION SOFTWARE

(a) Section drawing.

(b) Automatically generated meanlengths and strip size.

(c) Dimensioning.

(d) Paper size selection and scaling of drawing.

(e) Title block content printing.

2. THE FLOWER PATTERN SOFTWARE

(a) Flower pattern with common origin.

(b) Flower pattern with separate origin.

(c) Automatic bending radii control.

(d) Ability to process multiple element bending at the same stage.

3. THE 10 to 1 TEMPLATES SOFTWARE

(a) 10 to 1 template drawings for all forming stages according to the flower patterns.

(b) Automatic scaling down of template drawing sizes to fit paper width limit.

(c) Generation of template contour data output for further use.

(d) Percentage composite length definition for circular elements.

(e) Radii-sharpening option.

4. THE ROLL DESIGN SOFTWARE

(a) Automatic generation of top and bottom roll contour based on the template contour.

(b) Automatic incorporation of pinch-difference surfaces based on the supplied clearance values.

(c) Automatic separation of side-roll contour from the top and bottom roll contour when required.

(d) Automatic addition of extension-contours when required.

(e) Generation of roll drawing incorporating the selected optional features for all forming stages.

(f) Generation of roll contour data output for further use.

4.7.2 Software Input Requirements

1. THE FINISHED SECTION SOFTWARE

(a) Section geometry definition using linear and circular elements according to the given bending convention.

(b) Scale and paper size selection.

(c) Dimensioning selection.

(d) Selection of title block content print-out.

2. THE FLOWER PATTERN SOFTWARE

(a) Total number of bending stages.

(b) Details of element bending at each stage.

(c) Element length definition or monitoring at each stage.

(d) Radii-sharpening selection.

(e) Short-leg bending selection.

3. THE 10 to 1 TEMPLATES SOFTWARE

(As for the Flower Pattern Software).

4. THE ROLL DESIGN SOFTWARE

(As for the Flower Pattern Software, plus the following)

(a) Pass-heights and Centre-to-centre distances of top and bottom rolls.

(b) End tolerances of the intermediate sections.

(c) Selection of roll drawing output features.

(d) Selection of the desired forming stages for roll drawing output.

(e) Pinch-difference surfaces option definition data.

(f) Side-roll option definition data.

(g) Extension-contour option definition data.

4.8 Overall Software System Layout

According to the four distinctive functions of the roll

design activity, the overall software system had also been formulated into four separate program units for design and development purposes. Each program unit to a large extent operates independently so that selection of individual functions is possible when required. From the implementation point of view, separate program units require less overall computer core-store space for execution and from the development point of view, smaller units are more managable.

The layout of the overall software system is illustrated in Chart 4.1. The program execution is performed in two stages because of the intermediate data requirement. Typical outputs of the software are drawings as illustrated in Figs 4.4 to 4.8 and data files as illustrated in Appendix 1.

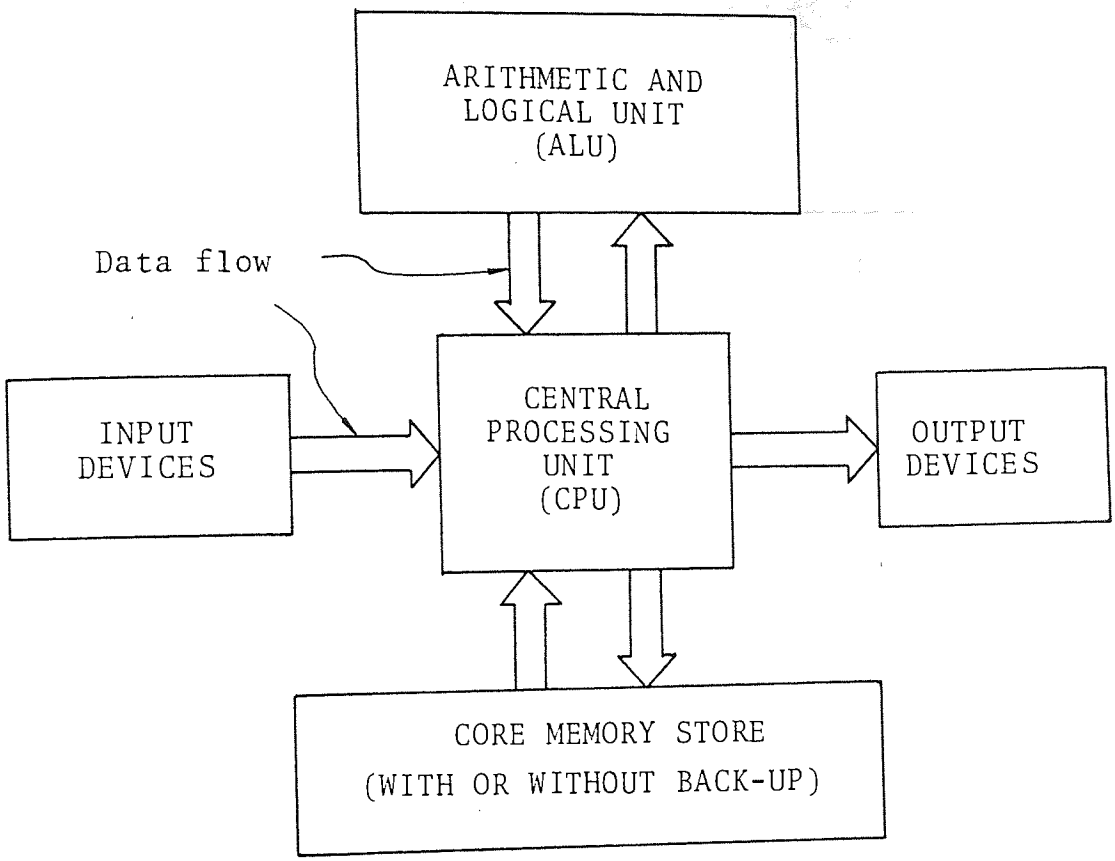


Fig 4.1 THE BASIC STRUCTURE OF A DIGITAL COMPUTER

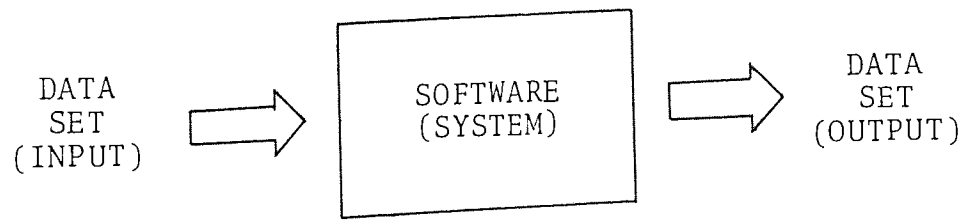


Fig 4.2 THE SOFTWARE PROCESS

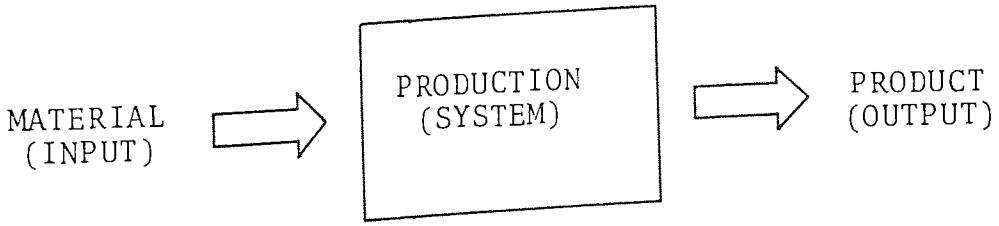


Fig 4.3 THE PRODUCTION PROCESS

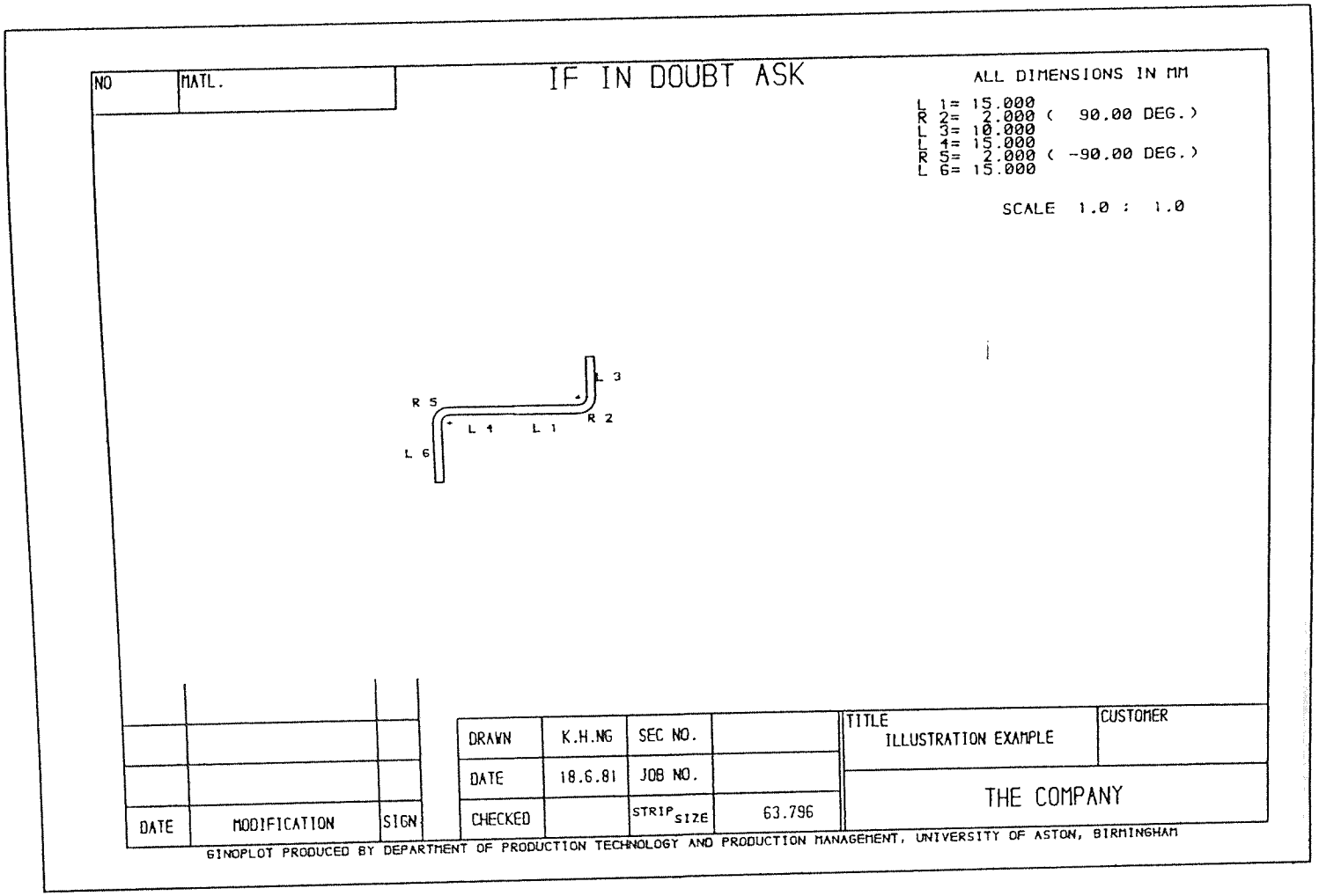


Fig 4.4 AN EXAMPLE OF THE COMPUTER GENERATED DRAWING OF THE FINISHED SECTION

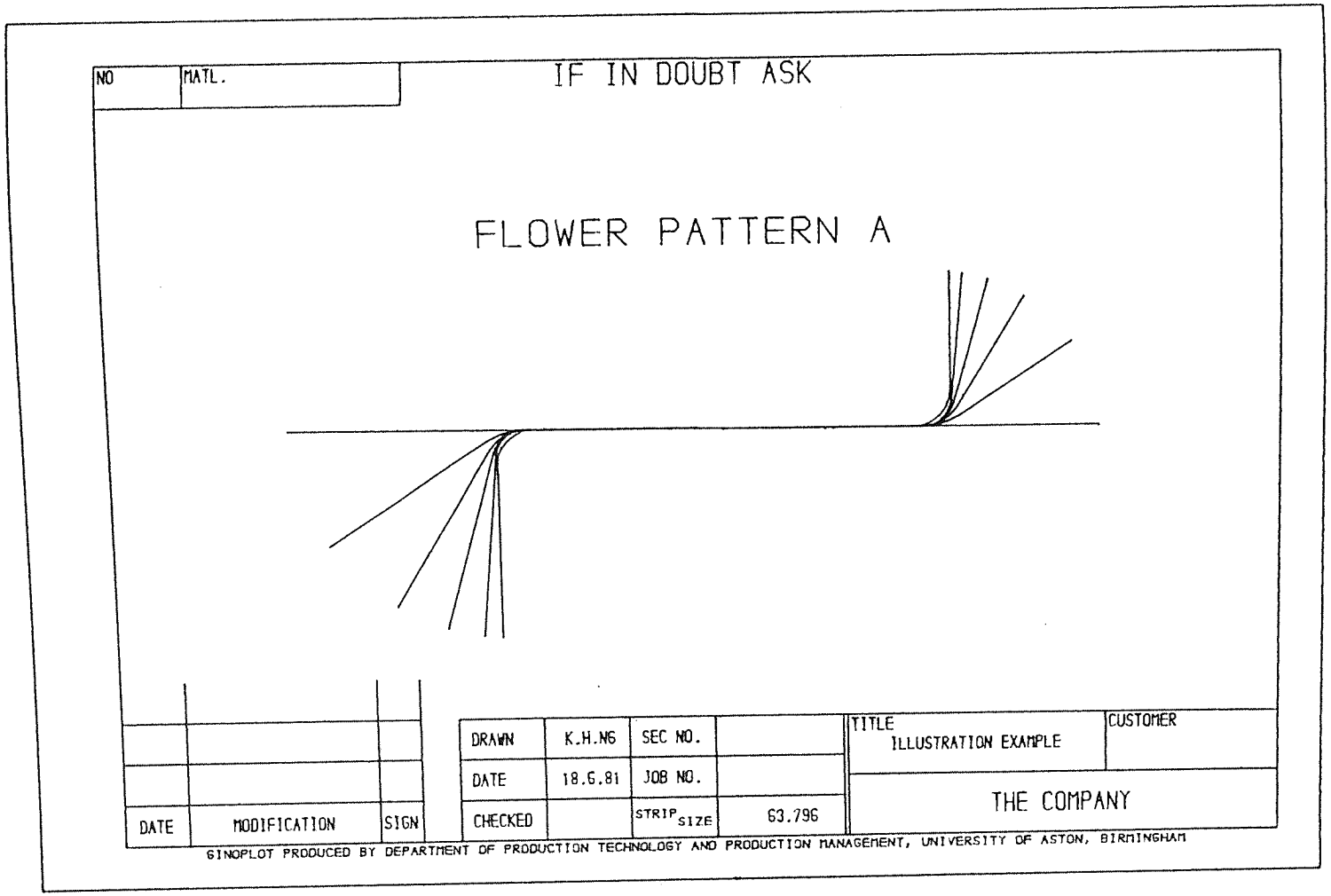


Fig 4.5 AN EXAMPLE OF THE COMPUTER GENERATED DRAWING OF THE FLOWER PATTERN WITH COMMON ORIGIN

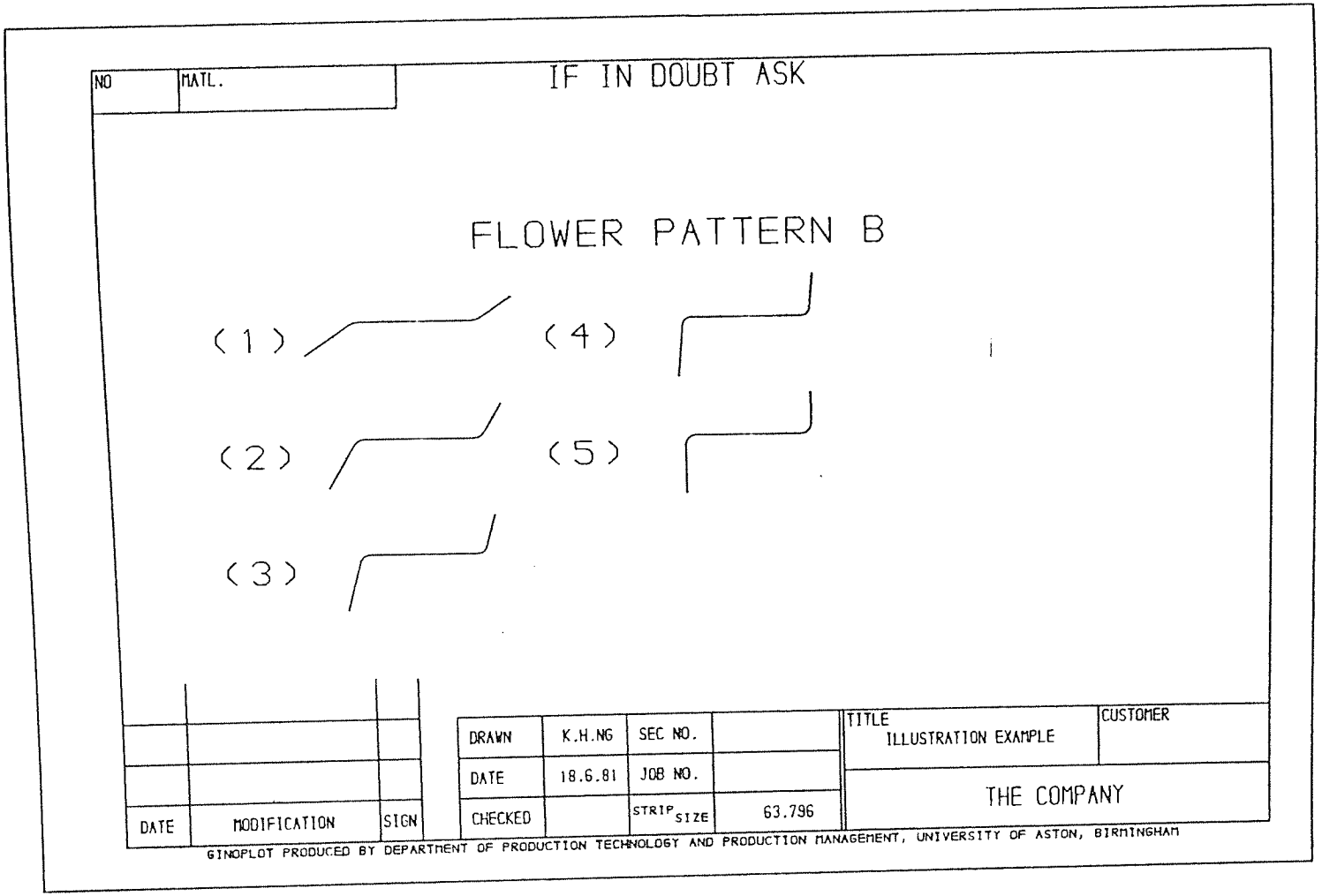
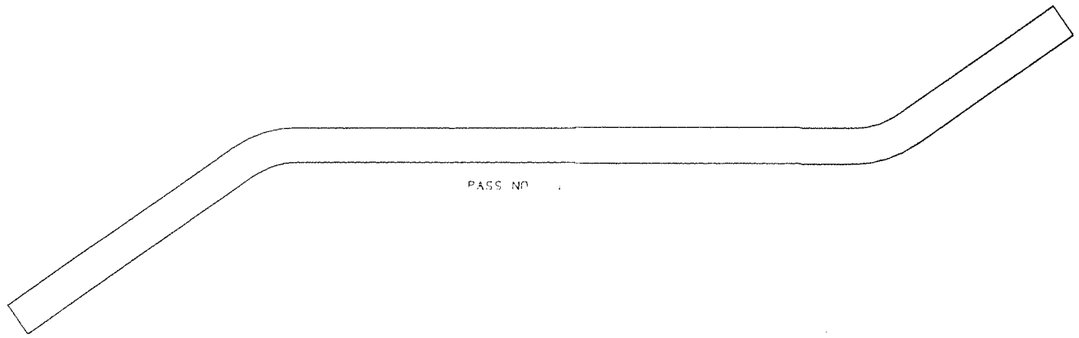


Fig 4.6 AN EXAMPLE OF THE COMPUTER GENERATED DRAWING OF THE FLOWER PATTERN WITH SEPARATE ORIGINS

STAGE NO.



PASS NO.

STAGE NO. 2
SCALE 10.0 : 1.0

PASS NO. 2

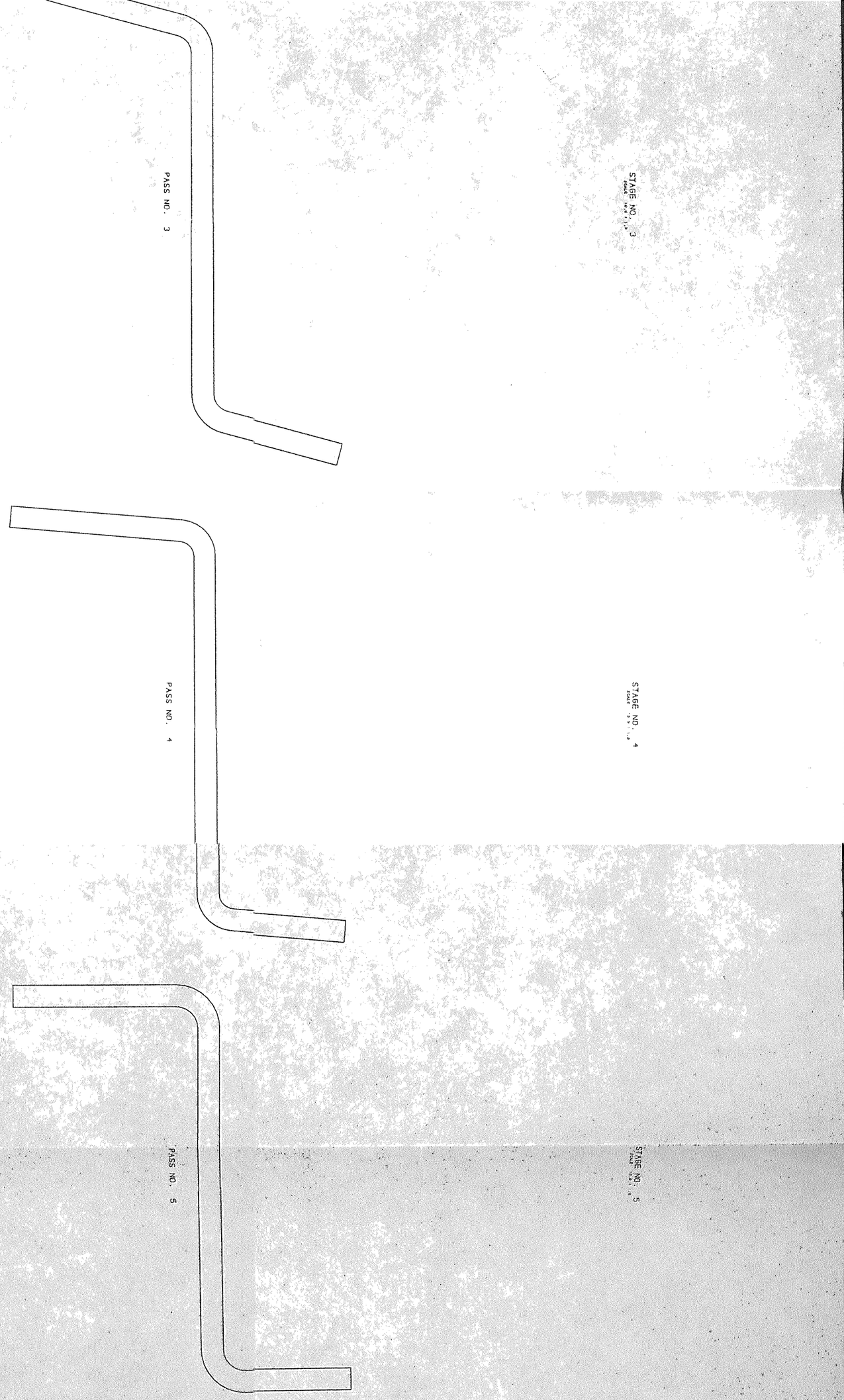
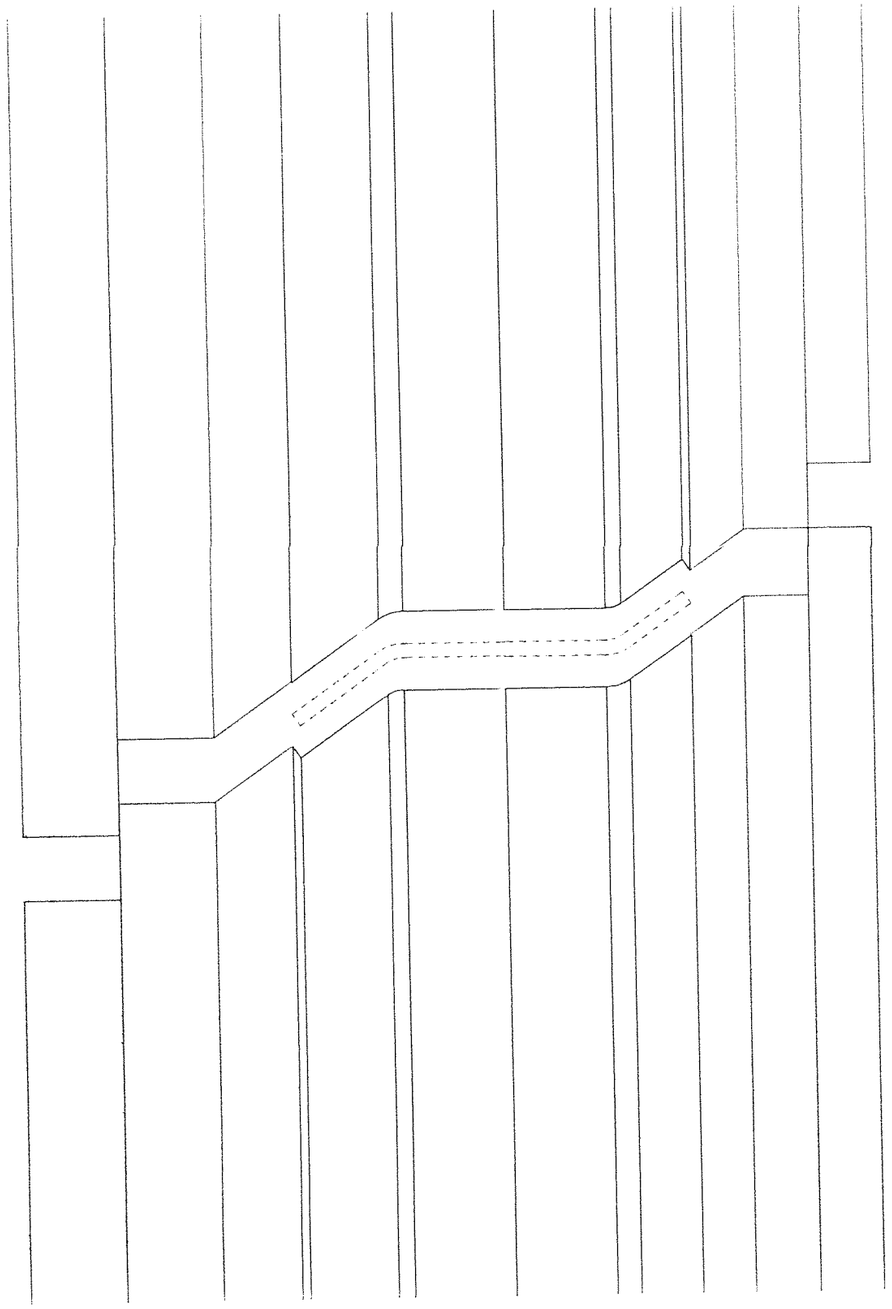
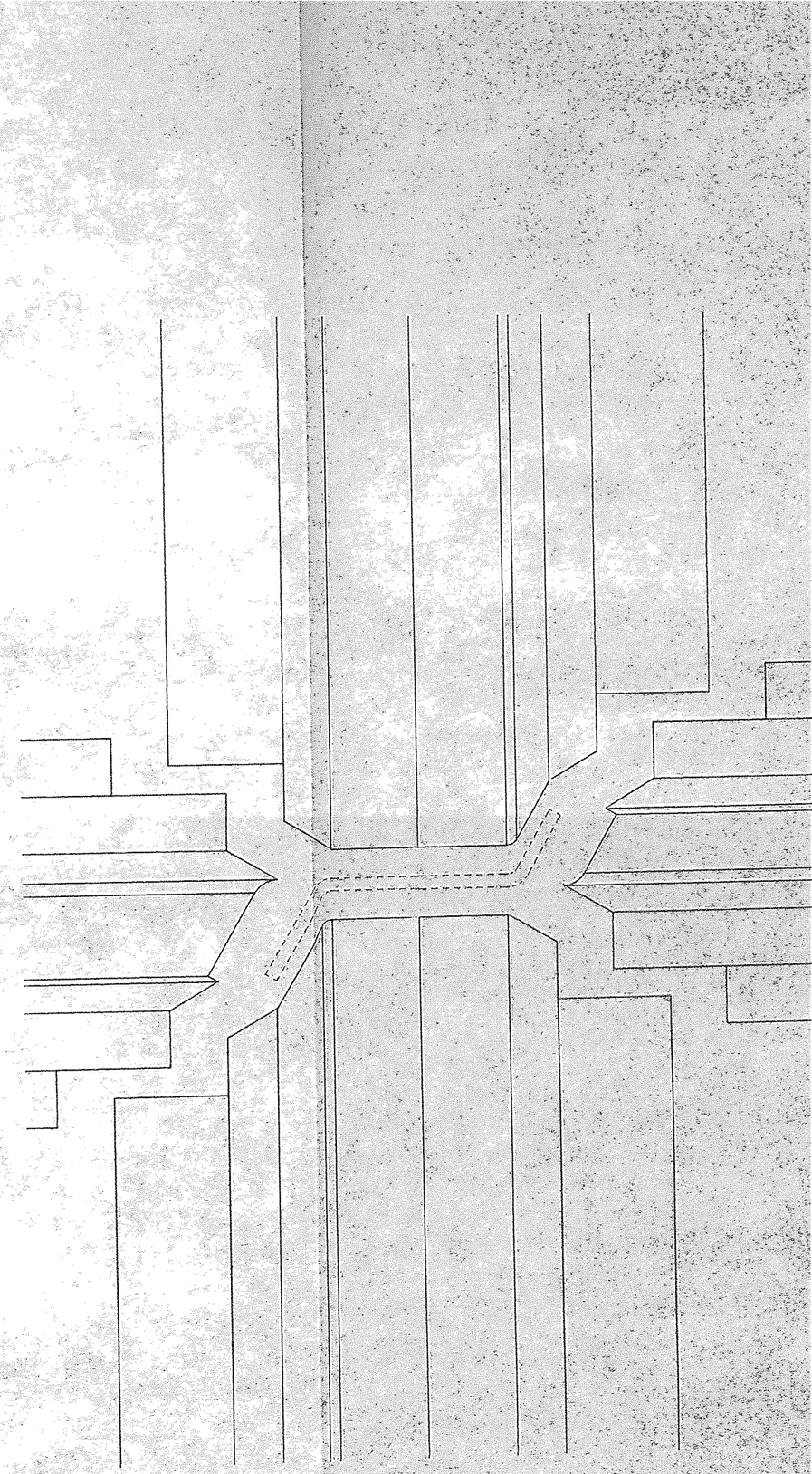


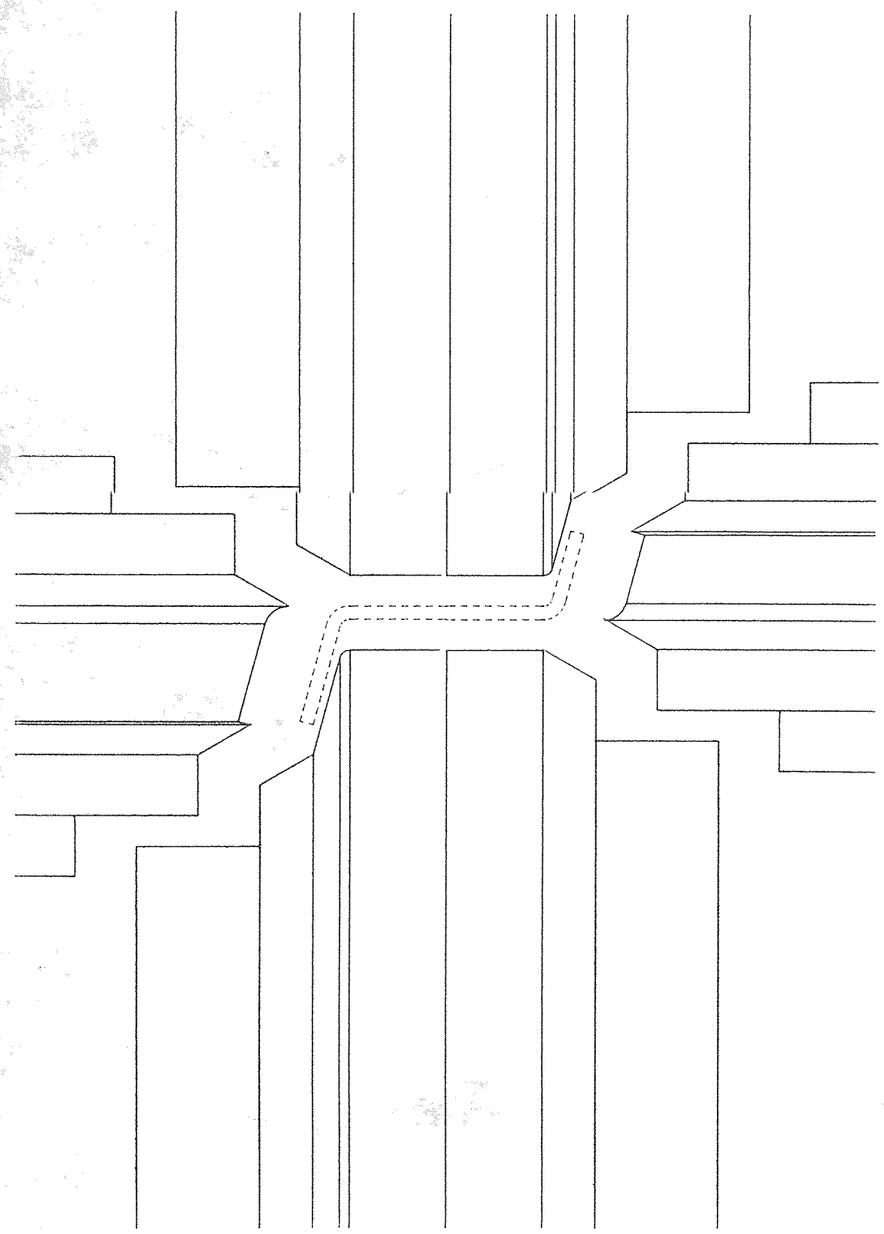
Fig 4.7 AN EXAMPLE OF THE COMPUTER GENERATED DRAWING OF THE 10 TO 1 TEMPLATES FOR ALL STAGES



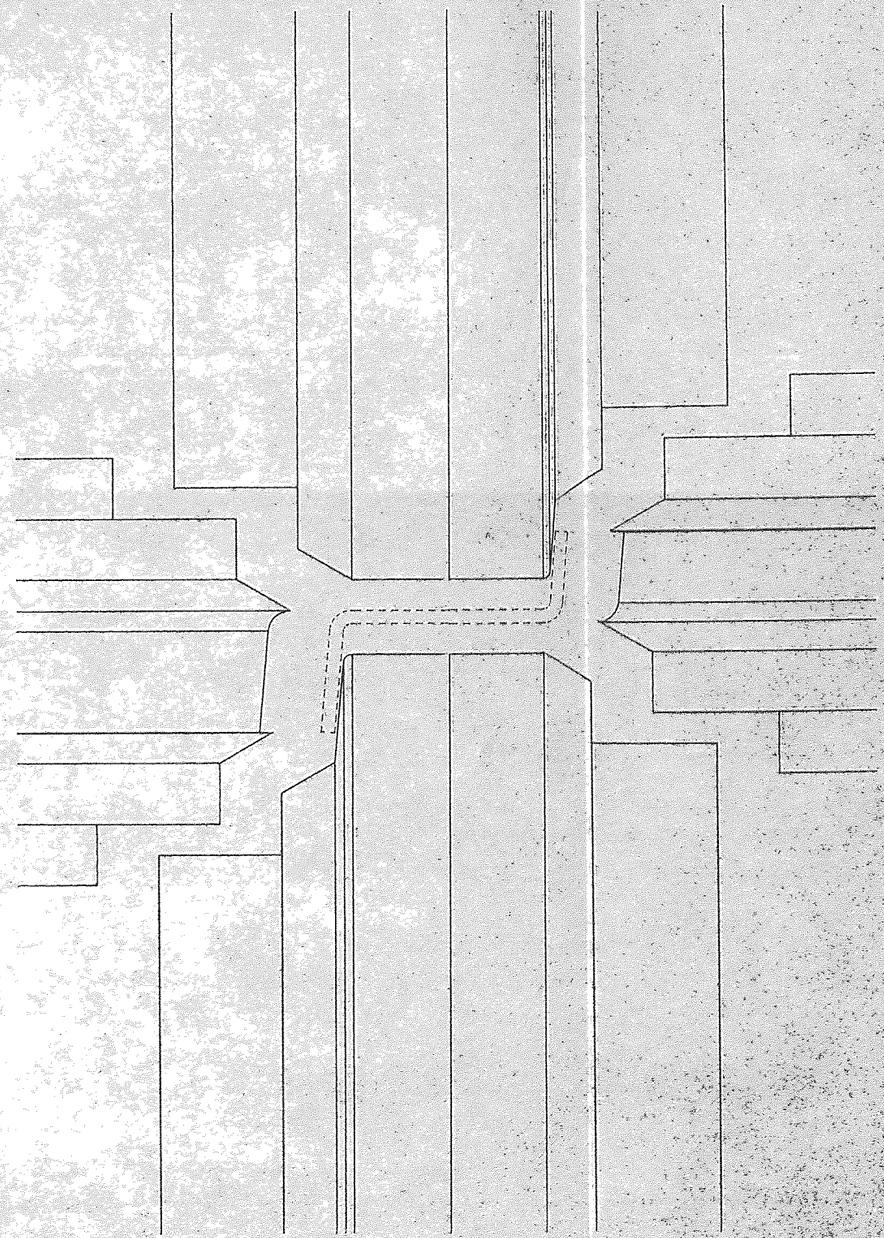
STAGE 1



STAGE 2



STAGE 3



STAGE 4

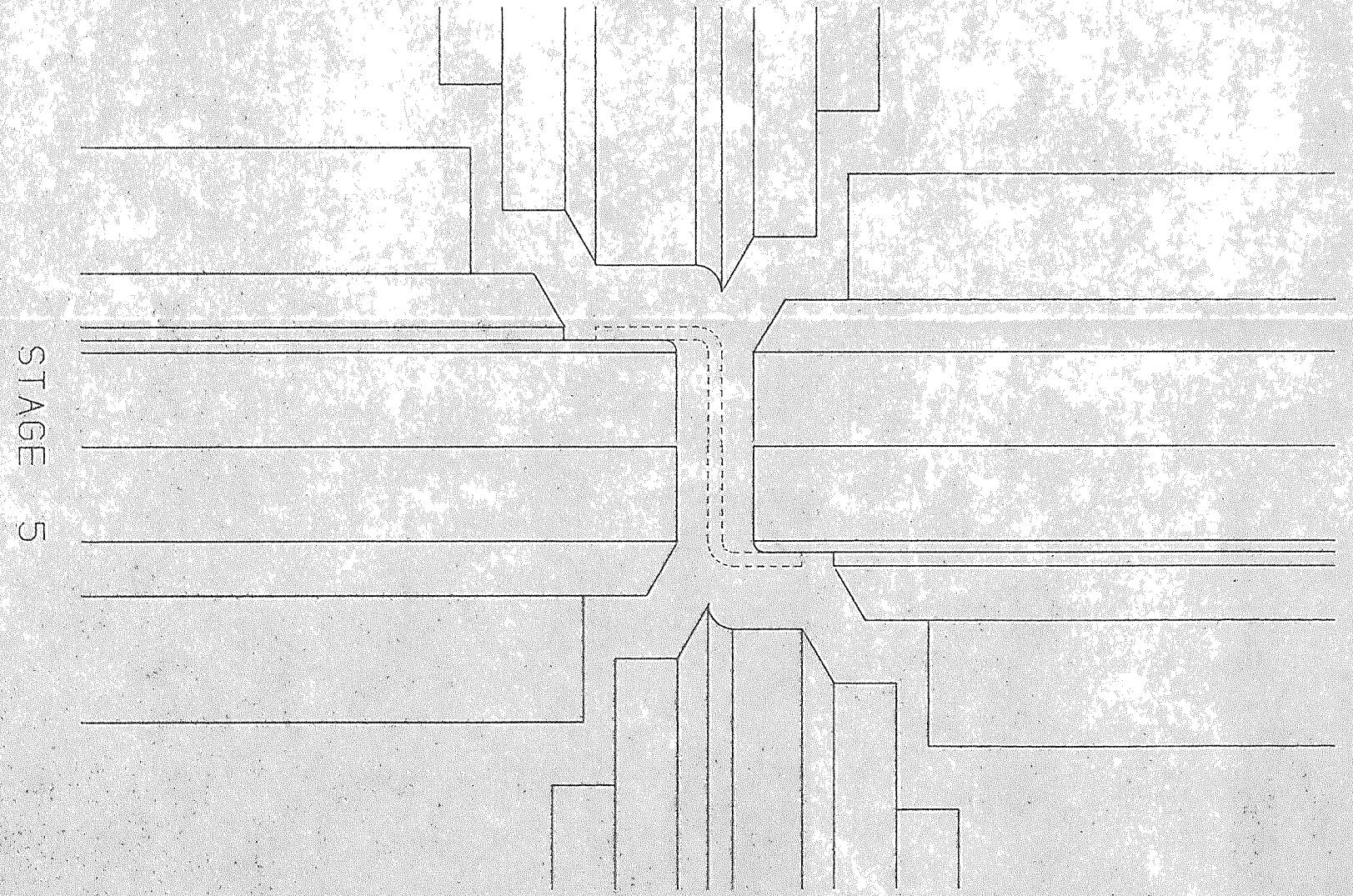
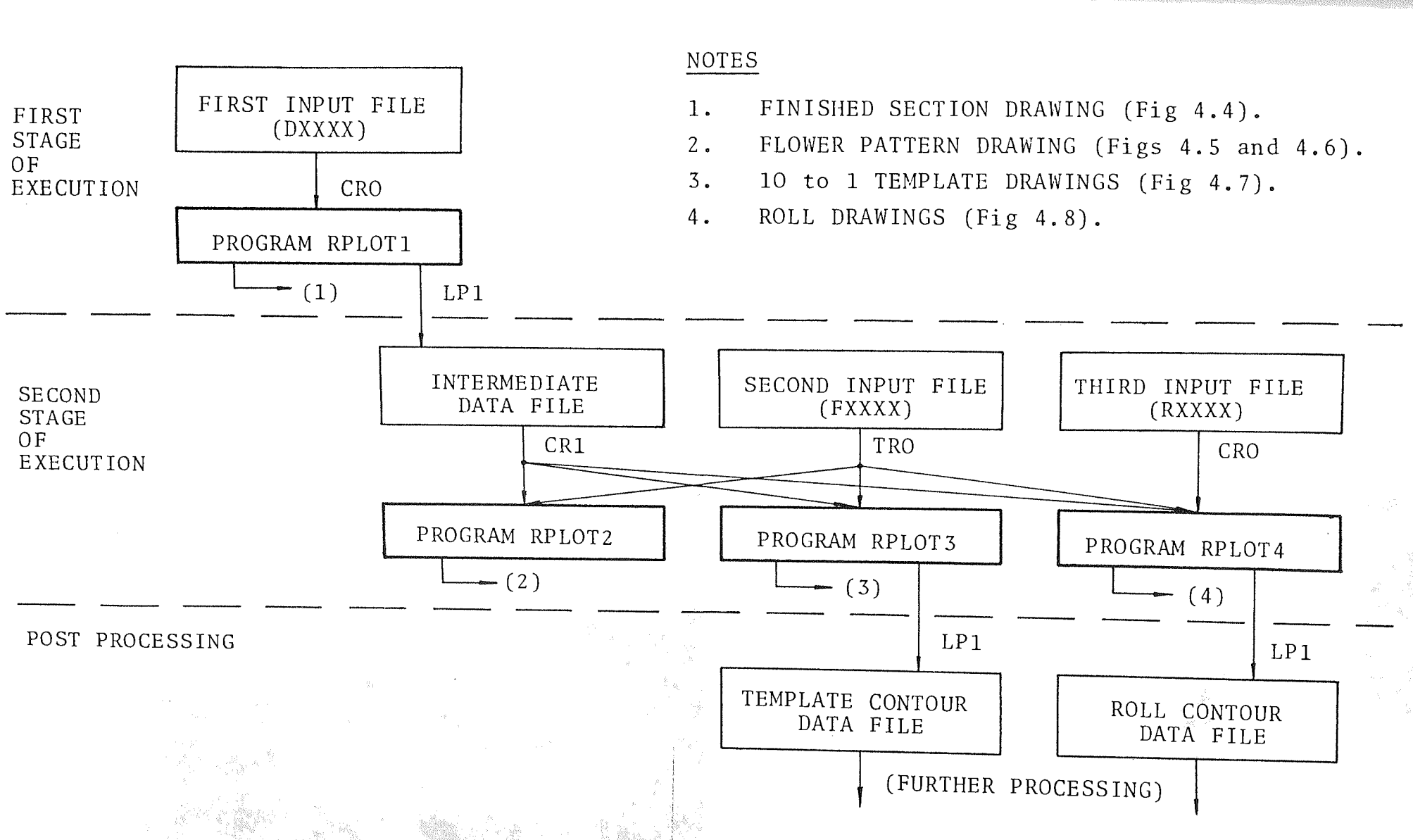


Fig 4.8 AN EXAMPLE OF THE COMPUTER GENERATED DRAWING OF THE ROLL DESIGNS FOR ALL STAGES



NOTES

1. FINISHED SECTION DRAWING (Fig 4.4).
2. FLOWER PATTERN DRAWING (Figs 4.5 and 4.6).
3. 10 to 1 TEMPLATE DRAWINGS (Fig 4.7).
4. ROLL DRAWINGS (Fig 4.8).

CHART 4.1 LAYOUT OF THE ENTIRE ROLL DESIGN SOFTWARE SYSTEM

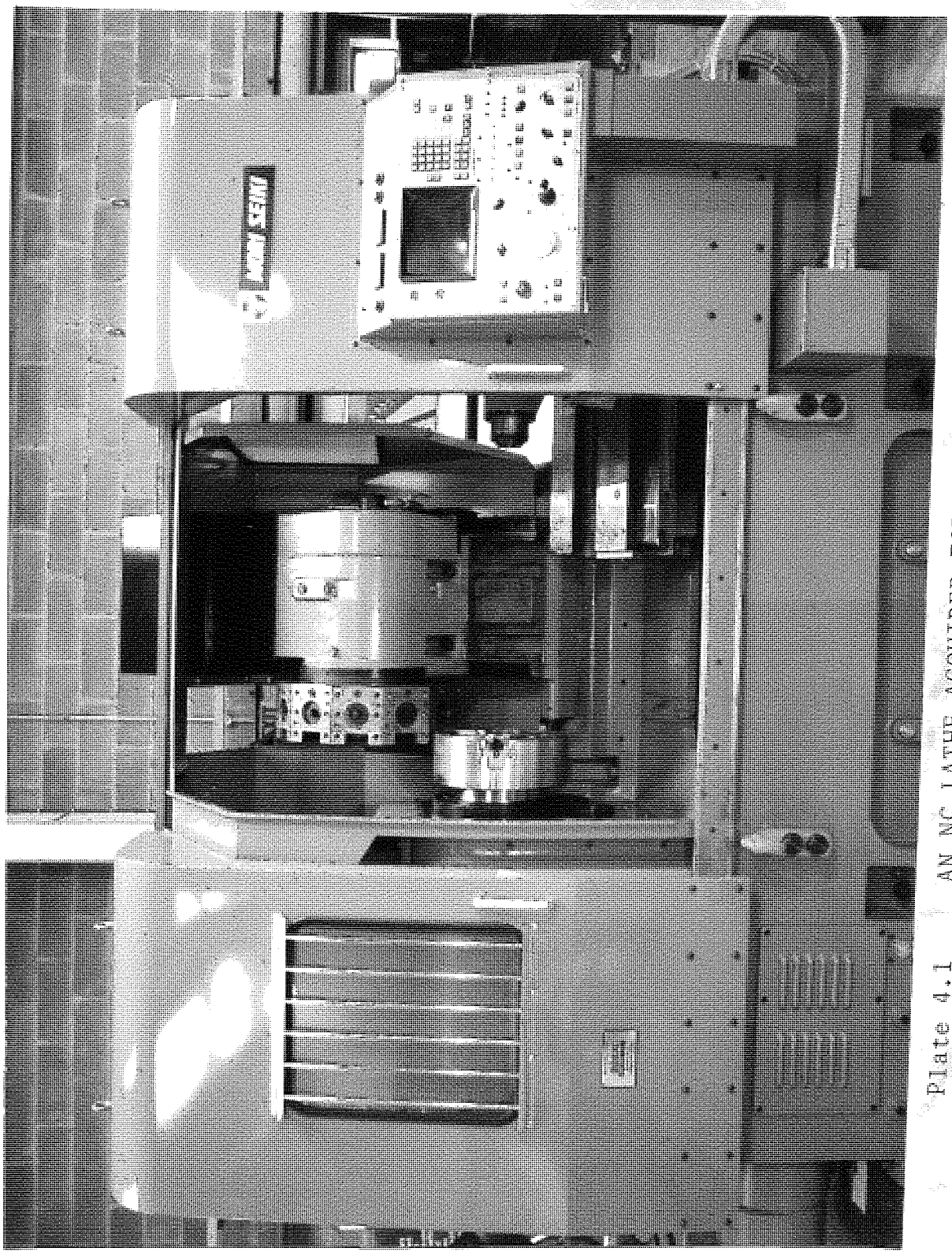


Plate 4.1 AN NC LATHE ACQUIRED FOR ROLL MANUFACTURING

CHAPTER 5

GENERAL APPROACH IN THE FORM-ROLL SOFTWARE DESIGN AND DEVELOPMENT

5.1 Introduction

Common lack of scientific approach for software design and development, coupled with frequent mismanagement in software projects, had in the past resulted in the creation of many unreliable and inflexible software products, often with heavy penalties[26]. In fact the history of software development in general is full of disasters which were costly and sometimes fatal[27, 28]. In recognition of the problems, many new software tools have since been introduced in an attempt to improve the situation, with limited success. Although these new tools are accepted to some extent in the software industry, they do not necessarily produce the claimed results when applied under non-identical circumstances. Furthermore, quite often even experts cannot agree among themselves about the right way to apply these tools. One typical example is Structured Programming[29], which has received wide attention, but the usage of which is still full of controversies[30]. Despite the confusions, these tools could still be helpful provided

they are selectively and cautiously applied. In this research, software design and development constitute a vital part in the whole computerisation programme. Systematic considerations of relevant factors affecting the software have therefore been carried out throughout this development and they will be discussed in detail in this chapter. Several more commonly used software tools related to this development will also be discussed.

5.2 Software Quality Characteristics

The description of software quality characteristics in current practices is still somewhat vague. In fact, different experts often have rather different ways of defining and classifying these characteristics[31]. General lack of standardised terminologies and difficulty in evaluating these quality characteristics accurately and objectively are among the chief causes for such a confusing situation. Software acceptance tests to date are still commonly conducted and assessed on a very subjective and often personal basis. The following is a list of the more commonly used characteristics, just as a guide:

5.2.1 Reliability

This is loosely used to describe a quality that acceptable results are always produced by the software. It implies ACCURACY and CONSISTENCY or REPEATABILITY of ACCURACY, of software output results, which in turn imply ACCURACY and COMPLETENESS of the software itself.

5.2.2 Portability

This is used to describe a quality that the software can be successfully implemented on different machines with ease. The term COMPATIBILITY is sometimes used to refer to the same thing but it may mean other things too. Purists demand that truely portable software must be transferrable from one machine to another without any modification. Although that is great for the sake of argument, it is rather idealistic and therefore could hardly be applied in practice. A more realistic interpretation of PORTABILITY should therefore be based upon the amount of effort required to tailor the existing software to suit its new operating environment. In some cases, program conversion may be necessary especially if the target system supports a totally different programming language.

5.2.3 Flexibility

This is used to describe a quality that existing software can be changed with ease. Change here may mean moving one part of the software to another, removal or addition of parts or alteration of the existing parts themselves, to suit new situations. FLEXIBILITY normally implies MODIFIABILITY and EXTENDABILITY (or sometimes AUGMENTABILITY) and should contribute to MAINTAINABILITY.

5.2.4 Clarity

This is used to describe a quality that the purpose, function and structure of the software can be easily understood. It implies easily recognizable and consistent program structure, efficient organisation and easily trackable logic sequence and data groupings. Complicated and obscure structures should be completely avoided if possible, or be kept to a minimum if they are unavoidable, for simplicity. It is generally accepted that CLARITY contributes towards UNDERSTANDABILITY, FLEXIBILITY, MAINTAINABILITY, TESTABILITY and ACCURACY. Structured Programming is often adopted to maximize CLARITY.

5.2.5 Maintainability

This is used to describe a quality that the software can be kept in satisfactory working condition with ease during its intended life of service. Usually it is viewed as a quality which becomes necessary as a result of lack of RELIABILITY. Sometimes, however, it is also interpreted as a quality for accommodating minor software changes due to changes in users' needs or the objective environment.

5.2.6 Testability

This is used to describe a quality that the software developed can be tested thoroughly with ease. Test cases are often designed according to performance requirements and program capabilities.

5.2.7 Efficiency

This is used to describe a quality that the software can be implemented with minimum (or optimum) use of resources. In the case of software implementation, it really means the use of least possible core-store space and shortest possible time for software execution on

the computer. Both core-size and timing are vital in REAL-TIME systems but less so in batch-processing systems. Efficient use of core-store space and execution time however is economically desirable. In general, EFFICIENCY is secondary in importance to RELIABILITY as without the latter it is meaningless. Modern solution for boosting software efficiency is the use of optimizing compilers, which are specially designed to translate source programs written in high level languages to efficient machine codes. The old way of using tricky codes to save a few lines of coding is regarded as highly undesirable practice from the maintenance point of view.

5.3 Common Software Design and Development Techniques

5.3.1 Modularity

This is a technique for dividing a piece of software into self-contained and discrete units. Its chief advantages are ease of software manipulation and structuring. Its chief disadvantage is that the number of data interfaces increases as the number of units or modules increases and inefficient software operation may result. The subprogram segment,

such as the SUBROUTINE subprogram of FORTRAN, can be considered as a typical simple module. A complex module may consist of several simple modules.

5.3.2 Structured Programming

This is a technique[28, 29] used to construct a program using only three basic logic structures as shown in Fig 5.1:

1. Sequence

   This is a structure for executing a succession of operations.

2. If-then-else

   This is a structure for executing mutually exclusive operations according to the associated condition.

3. Do-while

   This is a structure for executing an operation repeatedly while the associated condition holds.

The main advantage of Structured Programming is that it greatly improves program clarity. It is theoretically

possible to construct any program with only the three basic logic structures described, resulting in a program structure without GO TO statements and labels. The consequence of such extreme practices, however, is the need to duplicate logic sequences at different places of the program or to create an excessive number of subprograms, both of which can lead to problems. Complete elimination of GO TO statements is therefore viewed unfavourably in practice. With a programming language like FORTRAN there is a total absence of block statement facilities, elimination of GO TO statements unnecessarily can only lead to more confusions.

5.3.3 Top-Down Method

This is a method[26, 28, 32] used both for designing and developing software. Software structure is to be organised in a hierarchy manner, with all higher level parts developed and tested before proceeding to the lower level parts. Dummy stubs are used to represent parts in lower level while parts in higher level are being designed and tested. Some Top-Down enthusiasts even go as far as using a special block logic representation language for Top-Down programming. It is generally regarded as a good method at the planning stage but not

necessarily as good when it comes to detailed design of the modules, especially when the higher level modules rely on their subordinate modules to perform substantial parts of their intended functions. In general some simulated or predictable stub behaviour must be generated for effective testing. The main advantage of the Top-Down method is early system integration.

5.3.4 Bottom-Up Method

This method[26, 28, 32], like the Top-Down method, is also used for both software design and development. With this method, the organisation of software structure is also in a hierarchy of different levels of logic control but development is from bottom level upwards. All lower level modules are designed and tested first with simulated driver modules before proceeding to develop modules in the next higher level. Its main advantage is the possibility of developing different modules of the same level in parallel at an early stage thereby saving time. Its main drawback is late overall system integration, which may be disastrous if communications among different module designers during early stages of development have been imprecise or inadequate, resulting in late module modifications or redesign.

5.3.5 Program Verification Technique

This is a technique[28] which is used to prove mathematically that the logic and behaviour of a program is correct and therefore reliable. Although the idea is great in theory, its practicality is rather doubtful. The process of proving the correctness of a program very often ends up to be even more complicated and much longer than the program itself, which means the time and effort required could be alarmingly large. As the proving process is manual, its correctness is in turn subject to another proof-out since no human activity is always faultless, so to obtain a perfectly correct program by this technique could still be a fallacy. Some software tools, like Structured Programming for instance, rely to a substantial extent on mathematical proving as the final guarantee of the program correctness since by themselves alone, a wrong program may still result.

5.4 General Software Design Considerations

In this section, systematic considerations regarding the various desirable software quality characteristics and the use of existing software design techniques for

the design of the present software system against the various constraints will be comprehensively discussed under the relevant headings.

5.4.1 Design for Reliability and Testability

Of the described software quality characteristics, the prime concern of this research has been reliability of the developed software. Every effort has been made to test all the parts and functions of the software as thoroughly as practicable to ensure its satisfactory performance during service. To facilitate efficient and comprehensive testing, the software has been designed for its clarity using Structured Programming principles to a certain extent and using a hierarchy of modular program layers. Careful data structuring with precise data definitions and isolated data manipulations wherever possible have helped much to preserve individual data integrity by minimizing the chance of data corruptions by mistakes and by increasing the trackability of error-sources. Strict discipline in program coding style has also helped in both logic trackability and testability. More will be discussed in subsequent sections.

5.4.2 Design for Portability

Portability is considered an important quality characteristic in modern software design and development for several reasons. One of the reasons is that useful software normally has long life because users tend to stick to software that had given them confidence in the past. Very often a computer system may have already become obsolete and discontinued requiring replacement with a different system while the software implemented on it is still indispensible. If the software has not been portable, great costs will be incurred for converting it to suit the new operating environment. The other reason is that useful software tends to become popular but it is rarely the case that all prospective users will have identical computer systems installed at their premises. A good example is the NAG (Numerical Algorithms Group, Oxford) library of Numerical Software[33], which is popular in the sphere of academic and industrial research in this country. To overcome the problem a common version of the software which can be implemented in most or if possible all of the prospective systems is the best solution since the unattractive alternative will be to keep a special version of the software for each individual user system resulting

in enormous expenses for software storage, maintenance and update. Although it is not always possible to produce a universally portable software in current state of software practices, it is still possible to design the software in such a way that the effort of modifying the software for transference is minimized.

In this research, there is a high probability that the software developed will eventually be implemented on a different system (or perhaps systems) in future, thus portability is an important consideration and has to be incorporated as far as possible. The two basic areas of concern have been the choice of suitable programming language and the choice of the graphic software and hardware facilities for drawing purposes. Considerations have been based upon the software development requirements against the facilites available. At the University Computer Centre, the only suitable graphic facility available then was the GINO-F library[34] supplied by the Computer Aided Design Centre, Cambridge. The library is accessible by programs written in FORTRAN only. FORTRAN was also available at the University Computer Centre on the ICL 1904S computer system supplied by ICL (International Computers Limited)[35].

Although in theory one is free to choose any programming language and any graphic facilities one prefers without having to stick to the use of FORTRAN and GINO-F library, the choice still needs to be justified. Most current graphic software facilities available commercially are based on two types of calling or host languages, namely FORTRAN and BASIC. BASIC tends to be used in smaller systems[25] whereas FORTRAN tends to be used on larger systems[24]. From this fact, FORTRAN can be considered a very suitable language to use. Being a high level language, FORTRAN is also highly portable and in fact it is one of the most popularly supported scientific language used in most computer systems. Other languages like ALGOL 68-R, PASCAL and APL are not as popular. BASIC in many ways resembles FORTRAN and conversion from one to the other is relatively straightforward as compared to other languages. Since portability relies to a great extent on the availability of the programming language on other computer systems, the use of FORTRAN is perfectly justified.

The chief disadvantage of using FORTRAN however is that there has not been a rigid standard in existence for the language for a long time, despite its popularity. The

most widely referred to standard is the ANSI (American National Standards Institute) FORTRAN, 1966[36] which itself is an imprecise standard. The result is that almost every computer manufacturer has its own interpretation of the standard and also its own idea of what additional language features not catered for by the standard should be incorporated. Such practices unfortunately have led to the creation of many FORTRAN dialects, each having some incompatible features not acceptable to the other and the ICL FORTRAN is a typical example. Such problems can be solved to a large extent by using the non-standard features of the language as little as possible. For portability reasons therefore, only standard FORTRAN features have been used freely in the software developed while the non-standard features have been used only when really necessary. To accomplish that a strict discipline in both coding style and program organisation has been followed.

The new standard FORTRAN 77 language published by the ANSI in 1977[37] is an extension of the 1966 version (FORTRAN IV). However, FORTRAN 77 compilers are still not widely available and are not as extensively used as FORTRAN IV. In any case, programs written in FORTRAN IV can readily be converted to FORTRAN 77 if necessary, with

only minor modifications. With some computer systems, special programs may be available to perform the conversion automatically.

The choice of the GINO-F library on the other hand was also justified because of the need to use graphic facilities for development purposes and also because of the fact that no better graphic facilities were available then. For portability reasons again, only the essential functions of the GINO-F library have been used. Besides, it is possible to construct a software relaying interface to translate each used GINO-F function to the corresponding function of other graphic software which is accessible by FORTRAN. For the graphic software not accessible by FORTRAN, an intermediate file of graphic data in alphanumeric form may be used as the relaying interface. With the existing arrangement, namely the ICL FORTRAN with the GINO-F library, no such interface is required.

5.4.3 Design for Flexibility

Changes are often necessary at almost any stage of software development. Even when the developed software is already in service, constant updates may still be

required to ensure its satisfactory performance throughout its useful life. Built-in structural flexibility of the software to accommodate these likely changes is therefore vital since without it changes might be difficult and hence costly to carry out. In some cases software inflexibility may render the changes impossible thus requiring new software to be created in its place. From the maintenance point of view, unforeseen errors in the developed software may occur from time to time and sufficient room for manoeuvre should therefore be provided for remedying the imperfections. The two main aspects of software design which affect the software flexibility are program structuring and data structuring.

Modular design for software can be considered an effective way of enhancing the manipulatibility of the program structure. Normally the sequence of logic execution can more readily be reshuffled by altering the arrangement of the calling instructions to the corresponding modular subprograms than by shifting blocks of logic sequence from place to place. Logic sequence which is repeatedly used at different parts of the software can normally be put into a commonly accessible subprogram so that when changes are required in the logic sequence there is no danger of updating one part

and forgetting the other. Logic independence may be increased considerably by grouping closely related functions into the same module so that changes when occur will not spread to other parts of the software. By using single-entry and single-exit for each subprogram, complex logic branching patterns which are highly risky to modify may be reduced.

Data structures on the other hand can also be designed for ease of data manipulation and for flexibility. Arrays have been used extensively in the developed software for random data access purposes. Each data has its type, values and function strictly defined for its intended purpose to avoid confusions. For instance, a read-only data value once set up is strictly used only for read-only purposes even though the programming language feature permits unauthorised alteration of the data value. For the arrays used in the software spare elements have been included to accommodate unforeseen needs and changes. In addition, the arrays have also been designed open-ended for probable future extension purposes.

Other than program structuring and data structuring,

strict discipline in coding style has also been followed thereby increasing the clarity of the software and making changes easy to perform.

5.4.4 Input Data Validation

In computing, the input data set is a major factor which affects the outcome of the results, namely the output data set. The plain truth is that if mistakes were present in the input data set then the output data set could never be right, a situation described in the software industry as "garbage in, garbage out". In modern days, "garbage out" is no longer a laughing matter, as most experts agree its consequences can be very serious indeed, such as a matter of life or death, depending on what the computer involved is used for. To prevent the disappointing or sometimes tragic results from happening, strict control over the input data set is therefore necessary, at the very least effort should be made to reduce the chances of wrong input data being accepted. Throughout the software development in this research, input data validation measure has been applied wherever possible when dealing with the user supplied input data. Facilities to detect possible errors in the data, to echo-print the data and to display

appropriate messages to indicate the error nature have been incorporated in all input data processing functions.

5.5 Software Development Strategies

Apart from software design considerations, another importnat part of software development is the formulation of a suitable development procedure. As in software design, there is again a general difference of opinions in this respect. In general different situations would require quite different kinds of development techniques and procedures. Some of the commonly used development techniques described earlier have been applied selectively. Together with some of the author's own ideas, promising results and progress have been accomplished.

5.5.1 Modules Construction Sequence

Two widely adopted strategies for planning and implementing the module construction sequence are the Top-Down and the Bottom-Up methods as described earlier. Decomposition of a complex software system into modules which form a hierarchy of distinctive layers of logic control is the common feature used in both methods, but

the sequence of developing the modules in one is exactly opposite to that in the other. The Top-Down method starts with high level modules and proceeds downwards while the Bottom-Up method is just the opposite. Both methods have their own advantages and disadvantages when adopted and they both base the module construction sequence on the levels of modules in the hierarchy. In the author's opinion, although using the levels in the hierarchy is a convenient way of module separation, it is not necessarily a good basis for determining the module construction sequence. The reasons are that modules within the same level of control hierarchy do not necessarily relate to or communicate with one another and that temporary data and logic interfaces to simulate the behaviour of modules in the undeveloped level must be specially created for localised testing purposes. With either methods, changes or even redesign of the developed modules are more likely to occur since testing is rather localised and less integrated, especially when the modules are not rigidly or precisely defined. A more sensible way of developing is therefore to select a chain of modules which are closely related in the function they perform and develop them simultaneously. The chain of modules normally form a particular branch

of the overall hierarchy and the branch itself is in fact a miniature hierarchy. Within this branch, the Top-Down method may be used during the planning stage whereas the Bottom-Up method can then be used during the detailed design stage. This way permits the building up of the entire system hierarchy branch by branch instead of level by level and the result is a more integrated structure. If the branches are sufficiently independent of one another in both logic and data interaction, then parallel development can be implemented, otherwise the more independent branch should have to be developed first.

5.5.2 Data Flow Analysis Technique

This is the author's own technique which has been used quite successfully in the present software development work. With this technique, the decomposition of software structure is done according to the data flow patterns, enabling a hierarchy of different branches rather than levels of logic control to be constructed one by one. The theory involved are described below.

With complex software systems, the final output data set usually cannot be mapped directly to the initial input

data set. Depending upon the complexity of the functions involved, the number of intermediate states of the data set varies. Normally it is possible to decompose the overall function that operates on each particular data set into subfunctions that can be executed in series (Fig 5.2). Mathematically it can be shown that such decomposition is valid with respect to the accuracy of the results:

Let X_i be the input data set and X_o be the output data set and let X_1, X_2, X_3, X_4 .... X_n be the intermediate data sets.

Then the overall function or mapping process of the data may be represented by:

$$X_o = f(X_i) \qquad (5.1)$$

The function may then be decomposed into the following subfunctions:

$$X_1 = f_1(X_i)$$

$$X_2 = f_2(X_1)$$

$$X_3 = f_3(X_2)$$

$$
\begin{aligned}
X_4 &= f_4(X_3) \\
&\vdots \\
X_o &= f_{n+1}(X_n)
\end{aligned}
\qquad (5.2)
$$

If X_1 can be shown to be correct after thorough testing, based on sets of input data values X_i which are known to be correct, then the correctness of subfunction f_1 can be confidently confirmed. X_2, X_3, X_4 and so on until X_o can then be mapped stage by stage successively to X_i as development proceeds, during which the correctness of each subfunction f_2, f_3, f_4 ... f_{n+1} in turn can also be established one by one. It follows that if each subfunction f_1, f_2, f_3 ... f_{n+1} is correct, then the overall function f that the subfunctions collectively represent must also be correct and reliable, so is the output data set X_o given that X_i is correct.

In general it is possible to identify the main streams of data flow of a software system and to design suitable data structures to contain them so that each stream could be isolately identified, manipulated, tracked and examined throughout its flow. Independent streams

may be designed and developed in parallel while dependent streams can be developed after the completion of the related independent streams. Having streamed the system data flow, the overall function of each stream can then be decomposed into series of subfunctions which will then be designed and tested one by one in their natural sequence. The intermediate data set output by one subfunction can be used as input for the next subfunction in the same stream, as each subfunction of the stream is being developed in sequence. The initial input data set to the stream serves as a constant source of test data for proving out the reliability of each subfunction stage by stage until the last subfunction of the stream is completed. Special test interface design is thus not required with this arrangement since realistic test data are constantly flowing from the source of the stream through the developed subfunctions to the subfunction being developed. Testing is systematic as consistent checks on the content of each intermediate data set ensure timely and efficient detection of errors and their removal. Closely related data can always be grouped into a common stream for easy tracking and to minimize the chance of negligence. Consider the following mathematical illustration:

Supposed there is a function:

$$Z = f(X, Y, U)$$

Because of negligence, instead of taking into account the influence of all the relevant sources, some may have been left out by mistake, resulting in a wrong Z value. Proper data grouping can reduce such risks.

To perform the subfunctions of each data flow stream, complex modules corresponding to the subfunctions may be designed, forming various branches of the system software hierarchy. Detailed design of each subfunction module can then be carried out as described in the last section.

The Data Flow Analysis technique can in fact be used as a complement of Structured Programming to accomplish software reliability. The main weakness of Structured Programming is that it concentrates only on logic structure alone without taking into account the influence between data flow and the logic functions. As a result, a program structure which is designed completely according to Structured Programming principles is not necessarily a correct program, since there is still a strong possibility that unrelated data are operated wrongly in

a perfectly correct program structure.

Take for example the sequence structure, supposed an intended function Z = X + Y + U has to be split up and computed in parts, a wrong relationship may still occur with a perfect structure as shown in Fig 5.3, there is therefore no guarantee on obtaining reliable results. Although in theory such weakness may be remedied by program verification techniques as described earlier but such approach is unfortunately far from practicable. With the Data Flow Analysis technique, flow and behaviour of data at any stage can be clearly observed and monitored systematically and thoroughly and hence the chance of continued presence of errors due to mistakes is greatly reduced. This effectively takes care of the vital aspect which Structured Programming fails to handle.

5.5.3 Testing and Debugging Procedures

The purpose of testing and debugging is to ensure that each part of the software designed to perform the specified functions is correct and reliable. With program modularity, flexibility and clarity, logic and data trackability of the software is improved enormously, permitting more systematic and thorough testing and

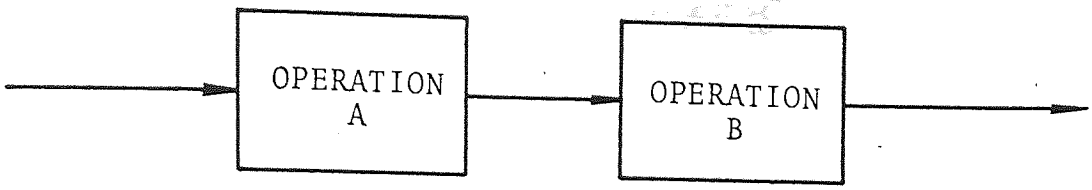
debugging. The decision to adopt the Data Flow Analysis technique and to determine module developing sequence function by function instead of level by level have helped to simplify the testing and debugging procedures tremendously. With each data stream, the source input data set is the only test data set required throughout its development since functions earlier in the stream are always developed before the succeeding one. Two methods were used to observe the behaviour of the intermediate data sets and to assess their correctness. One was the use of line-printer listing of their contents and the other was to represent their contents in the form of drawings on a plotter. Test cases were specially designed to stretch the capability of each designed function to its known limits. Each function was thoroughly tested before moving on to design and test the subsequent function in the same stream. Independent data streams had their functions designed and developed separately from one another in parallel while dependent data streams were developed only when their source input data sets, normally some intermediate data sets of the independent streams, were ready. By developing the functions of each stream in series, functions earlier in each stream were in fact exercised more frequently than their later counterparts since every test data supplied

at the input source had to pass through them before going to the function that was being developed and tested. That helped to ensure greater reliability of earlier functions in each stream. In general, it is more advisable to finish testing one function before starting to develop the subsequent function in the stream because unpredictable changes to the function may be required if mistakes are discovered in it and have to be corrected, probably requiring the redesign of the subsequent function as the one designed prematurely is no longer suitable. However, such risks at times may be necessary especially when time is precious. Such in fact was the situation in this development, mainly because the turnover rate of job processing at the University Computer Centre was painfully slow due to frequent overloading and breakdowns, making it necessary to attempt developing strictly series functions in parallel in order to make full utilisation of time. The outcome of such attempts varied according to how well the behaviour of each function turned out as expected.

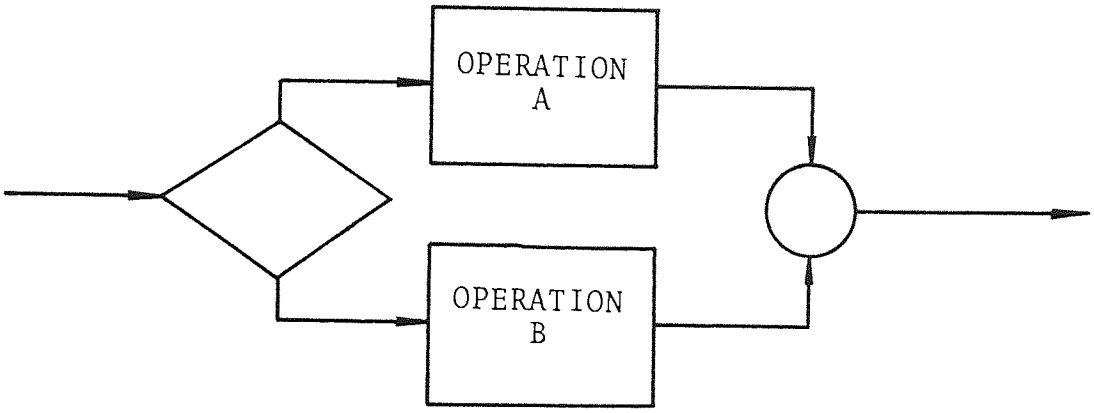
Preventative measures were taken as far as possible to minimize the chance of errors occurring. Programs had been proof-read thoroughly after coding before they were submitted for test runs. In this case, the ICL FORTRAN

compiler did help a great deal in generating relatively precise error-messages during compilation runs. On the other hand, error-messages generated during execution runs were unfortunately less helpful as they were less precise and less informative. Facilities for automatic logic tracing that could be triggered by program failure during execution runs were available and used, but other debugging aids like core-dumping and cross-reference symbol tables were not available. Generation of diagnostic print-outs of key data values by planting WRITE statements at appropriate points of execution was used extensively and constantly for debugging purposes, especially when other means failed to help. Error tracing was conducted systematically by back-tracking the data flow function by function until the error source was located. Modifications to the existing codes were done in the developed modules with careful considerations about their implications on other data and functions of the software as a whole. More obvious mistakes were usually detected during compilation runs and by visual inspection of the program codes. New modular subprograms were compiled on their own with a dummy main program for preliminary debugging before incorporating them into the main body of software being built up and tested, that had helped to remove most of the compilation errors quickly.

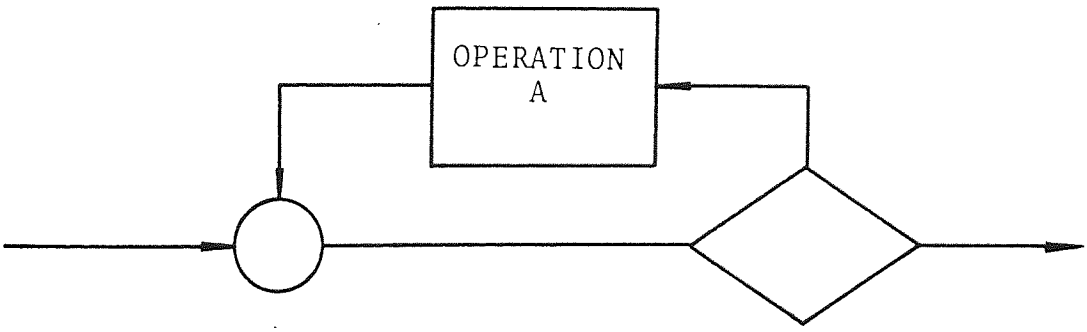
Errors which were the most difficult to trace and correct were the more subtle and random errors found in the intermediate data sets as revealed by drawings representing the data, they were mainly caused by genuine mistakes or inadequacy in the designed functions. Faulty functions were normally modified or occasionally even redesigned and retested until they work satisfactorily.



(1) SEQUENCE



(2) IF-THEN-ELSE



(3) DO-WHILE

Fig 5.1 BASIC LOGIC STRUCTURES IN STRUCTURED PROGRAMMING

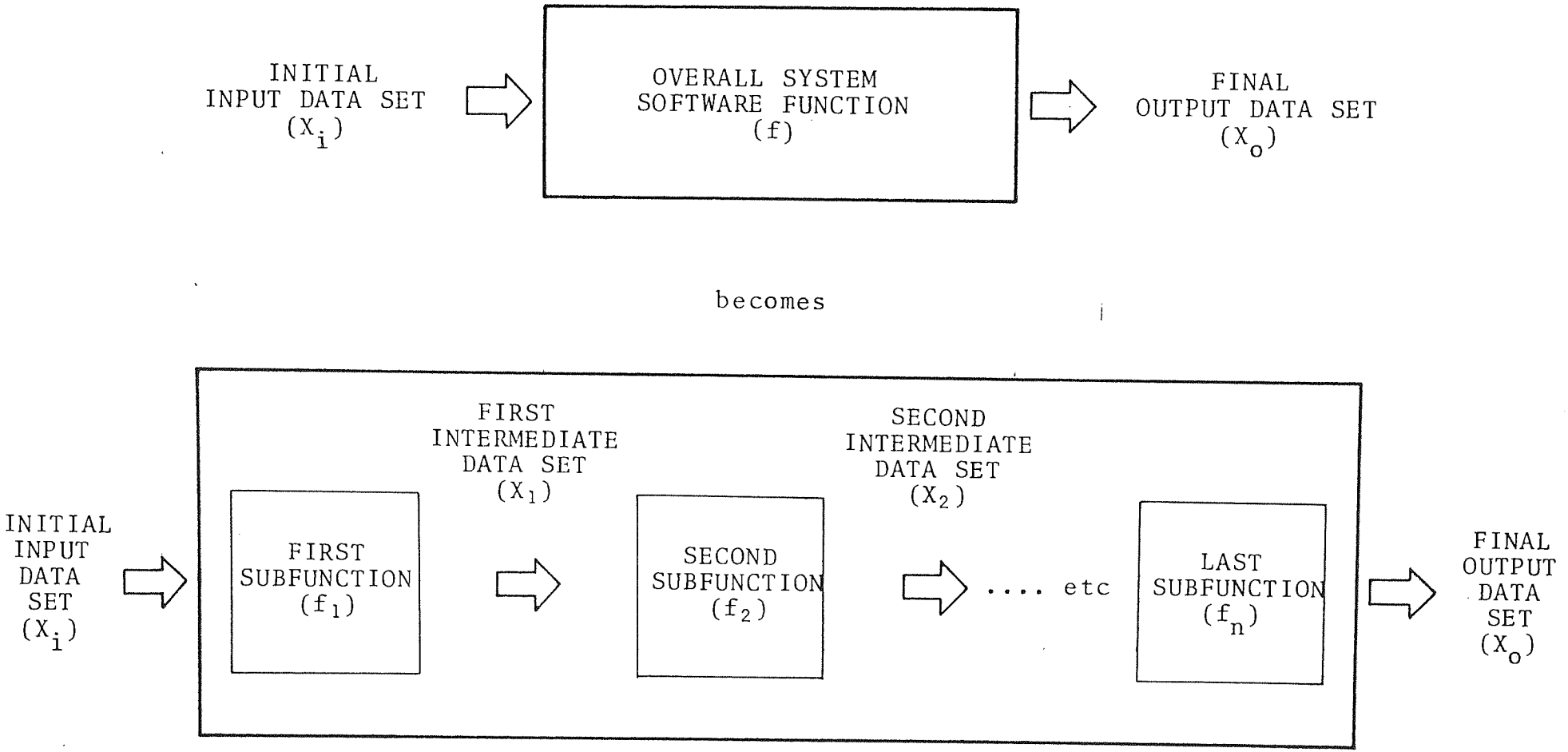
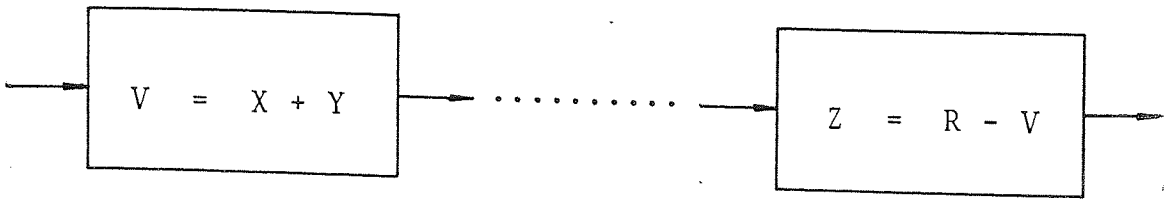


Fig 5.2 A TYPICAL DATA STREAM USING DATA FLOW ANALYSIS TECHNIQUE



Somewhere in front

Somewhere behind

(The required function: Z = X + Y + U)

Fig 5.3 POSSIBILITY OF MISTAKES IN PERFECT PROGRAM STRUCTURE USING STRUCTURED PROGRAMMING ALONE.

CHAPTER 6

THE FINISHED SECTION SOFTWARE

6.1 Introduction

The finished section (see Fig 4.4) is the ultimate product that the form-rolls are designed to manufacture, therefore its geometry must be accurately and adequately defined. A section definition scheme has been designed specially for this purpose, enabling users to describe the section geometry unambiguously and conveniently. Based on the definition data supplied by users, the finished section drawing can then be automatically produced to the desired requirements. Facilities for the selection of paper size, scale and dimensioning of the drawing have been provided. Individual meanlengths of all elements that form the finished section as well as the required strip size are automatically calculated. An intermediate data file is generated to retain data which will be needed in later stages of processing. Each of the abovementioned aspects of the finished section software will be discussed in detail throughout this chapter.

6.2 Section Definition Scheme

The sole purpose of the section definition scheme is to help users to define a wide range of section shapes accurately and without ambiguity in a convenient way. For the benefit of the users, it has been designed to bear close resemblance to the existing design practices.

6.2.1 Analysis of Section Geometry

The type of section geometry normally encountered in practice consists of only linear and circular elements with uniform thickness (Fig 6.1). Non-linear type of elements other than circular elements are normally not used but if used they may be approximated with circular elements. A section definition scheme that handles linear and circular elements is therefore adequate for the present purpose.

6.2.2 Thickness of Section

The raw materials used in Cold Roll-Forming are uniform in thickness and so are the sections produced from them. Deformation in Cold Roll-Forming normally occurs at the circular elements of the section, causing them to bend

progressively until the required angles of bend are obtained. It is therefore quite different from deformation in cold rolling which is solely aimed at reducing the thickness of material. For definition purposes, therefore only one thickness value needs to be specified for all elements of the section.

6.2.3 Definition of Linear Elements

A linear element may be represented geometrically by a rectangular block (Fig 6.2) with length l and width T, where T is the uniform thickness of the material or the section. In Cold Roll-Forming, linear elements remain linear and undeformed throughout all the roll-forming stages. Although it is not strictly true that the dimensions of linear elements remain absolutely constant during forming, the changes are usually not significant, especially with thinner sections. Provisions for element length adjustments (see Section 8.2) can always be incorporated to compensate for minor material flow in between circular and linear elements as a result of the rolling action, which is more likely to occur with thicker sections.

Provisions have been given for the use of linear

elements of zero length. These elements are necessary for several reasons. One reason is that the first element of the section definition sequence has to be linear so that occasionally inclination of the section may be defined when required. The other reason is that as a result of the inclusion of the composite percentage length definition scheme (see Section 8.3) circular elements cannot be defined in succession since during intermediate forming stage, certain parts have to be kept linear and linear elements of zero length are therefore necessary to separate the successive circular elements. Apart from that, they are also needed as dummy elements for composite circular element definition in the use of side-rolls (see Section 6.2.4).

6.2.4 Definition of Circular Elements

To completely define a circular element, with section thickness already defined, the users need only to specify its internal radius r and the final angle of bend A (Fig 6.3). Circular elements of zero radius, like linear elements of zero length, are peculiar elements occasionally needed for special usage. For instance, they may be used to form a compact section as illustrated in Fig 6.4 when bent

to 180 degrees.

Bending angles normally can be of any magnitude below 270 degrees provided the material bent does not interfere with the material of the section. For angles above 270 degrees there is a possibility that such interference may occur as illustrated in Fig 6.5. A check for such interference has thus been incorporated, using the following equation that accounts for the combined effect of radius, thickness and angular values:-

$$A_{max} = \sin^{-1} (r / (r + t)) + 270^0 \times \frac{\pi}{180} \qquad (6.1)$$

where A_{max} is the maximum permissible magnitude of bending (in radians),

r is the inside radius of the bend

t is the thickness of the section.

It is possible to define a circular part of the section using several elements (Fig 6.6). This composite form of definition is necessary when it is required to specify the break-point that separates top and bottom rolls from side-rolls (see Section 9.4.2).

6.2.5 Convention for Directions of Bending

The convention adopted for specifying the directions of bending at the circular elements is as follows:

Regardless of whether it is a left-hand bend or a right-hand bend, positive angles should be used for upward bends while negative angles should be used for downward bends (Fig 6.7).

This convention conforms to the usual practice of roll design and is easy to use.

6.2.6 Selection of the Type of Definition Sequence

It was found that to cater for all sorts of common section geometry with different numbers and combinations of circular and linear elements, five different types of definition sequence would be adequate.

Users are first required to choose the type of definition sequence by specifying the starting-point or point of reference (ORIGIN) type, from that point the sequence of element definition will begin. One of the following five types of definition sequence should be

selected:

ORIGIN = 1 (Starting from the left of the section and ending at the right).

ORIGIN = 2 (Starting from the right of the section and ending at the left).

ORIGIN = 3 (Starting from the middle of the section, towards the right and then from the middle towards the left).

ORIGIN = 4 (Starting from the middle of the section, towards the left and then from the middle towards the right).

ORIGIN = 5 (Starting from the middle of the section, towards the right and the section is symmetrical).

Certain restrictions have to be imposed upon the manner the elements of the section are defined with respect to the selected definition sequence type, because of the limit of complexity the developed software could have. For instance, in the case of sequence type 1

and type 2, inclination of the first element of the sequence, usually a linear element, is permitted within the range -90 degrees to +90 degrees only. Another restriction is that none of the elements defined in the sequence should cross the vertical line that passes through the point of reference (ORIGIN), otherwise errors will result during the later stages of processing in roll design.

6.2.7 Input Format for Element Definition

Four pieces of information, which together form a definition statement, are associated with each element in the definition sequence. They are as follows:

1. Element Number

   This is used as the identifier of each element in each definition sequence. A unique integer value within the range of 1 to 50 is assigned to each element of the sequence in ascending order.

2. Element Type

   This is used to indicate whether the element is of type linear or circular, 1 for linear and 2 for circular.

3. Element Length or Inside Radius

For a linear element, its length should be supplied whereas for a circular element its radius is to be supplied instead.

4. Bending Angle

This applies only to the circular elements and the sign is according to the convention for bending directions described earlier. Normally it is not applicable to linear elements except for element 1, in which case the angle is treated as angle of inclination to the horizontal plane.

For double-sided definition sequence types (ORIGIN is 3 or 4), two definition sequences are required for the left-hand side and the right-hand side of the section, instead of only one.

6.3 Computation of Section Geometry

The geometry of the section is represented in two parts, the top face contour and the bottom face contour. Each contour is made up of a series of element contours in the form of straight lines or circular arcs. They

are stored in arrays which are easily recognizable. The following describes the methods of computing the relevant geometrical data of the element contours.

6.3.1 Computing Cumulative Angle of Inclination

The cumulative angle technique is adopted to keep track of the orientation of each element contour with respect to the other element contours in the sequence. The cumulative angle of bending for each element contour is derived using the following equation:

$$\theta_{m+1} = \theta_m + \theta \qquad (6.2)$$

Where θ_{m+1} is the cumulative angle or the absolute angle of inclination of the current element

θ_m is the cumulative angle of the preceding element in the same definition sequence

θ is the angle of bending of the current element.

If element number is 1, then

$$\theta_{m+1} = \theta_m = \theta \qquad (6.2a)$$

Since θ is normally zero for linear elements except for element 1, it is therefore used to specify the angle of inclination for the section when required.

6.3.2 Computation of Linear Element Contour

In Fig 6.8, line P_1P_2 is the bottom face element contour while line P_3P_4 is the top face element contour. If the coordinates of the points P_1, P_2, P_3 and P_4 are known, the contours are defined.

The coordinates of point P_1 are available from the preceding element in the same sequence, so is the cumulative angle θ_m. With known values of the element length 1 and thickness t, the required points may be computed using the following equations:

For points $P_2(X_2, Y_2)$ and $P_3(X_3, Y_3)$ knowing $P_1(X_1, Y_1)$,

$$X_2 = X_1 + 1 \cos \theta_m \qquad (6.3)$$

$$Y_2 = X_1 + 1 \sin \theta_m \qquad (6.4)$$

$$X_3 = X_1 + t \cos (\theta_m + 90^0) \qquad (6.5)$$

$$Y_3 = Y_1 + t \sin (\theta_m + 90^0) \qquad (6.6)$$

For point $P_4(X_4, Y_4)$,

$$X_4 = X_3 + 1 \cos \theta_m \quad (6.7)$$

$$Y_4 = Y_3 + 1 \sin \theta_m \quad (6.8)$$

Or

$$X_4 = X_2 + t \cos (\theta_m + 90^0) \quad (6.9)$$

$$Y_4 = Y_2 + t \sin (\theta_m + 90^0) \quad (6.10)$$

The cumulative angle value θ_{m+1} for the current element is equal to θ_m since the element is linear and point P_2 will be the point P_1 of the next element in the same sequence.

6.3.3 Computation of Circular Element Contour

As illustrated in Fig 6.9 (a) and (b), there are two cases to consider in computing the geometry of a circular element using the known point P_1 and the known cumulative angles θ_m and θ_{m+1} as a result of the two possible directions of bending.

In the case of the upward bend (θ is positive), the

following sets of equations applies:

For the centre point $P_c(X_c, Y_c)$,

$$X_c = X_1 + (r + t) \cos (\theta_m + 90^0) \quad (6.11)$$

$$Y_c = Y_1 + (r + t) \sin (\theta_m + 90^0) \quad (6.12)$$

For the bottom face contour end-point $P_2(X_2, Y_2)$,

$$X_2 = X_c + (r + t) \cos (\theta_{m+1} - 90^0) \quad (6.13)$$

$$Y_2 = Y_c + (r + t) \sin (\theta_{m+1} - 90^0) \quad (6.14)$$

For the top face contour start-point $P_3(X_3, Y_3)$,

$$X_3 = X_1 + t \cos (\theta_m + 90^0) \quad (6.15)$$

$$Y_3 = Y_1 + t \sin (\theta_m + 90^0) \quad (6.16)$$

For the top face contour end-point $P_4(X_4, Y_4)$,

$$X_4 = X_2 + t \cos (\theta_{m+1} + 90^0) \quad (6.17)$$

$$Y_4 = Y_2 + t \sin (\theta_{m+1} + 90^0) \quad (6.18)$$

In the case of the downward bend (θ is negative), the following sets of equations applies:

For the centre-point $P_c(X_c, Y_c)$,

$$X_c = X_1 + r \cos (\theta_m - 90^0) \quad (6.19)$$

$$Y_c = Y_1 + r \sin (\theta_m - 90^0) \quad (6.20)$$

For the bottom face contour end-point $P_2(X_2, Y_2)$,

$$X_2 = X_c + r \cos (\theta_{m+1} + 90^0) \quad (6.21)$$

$$Y_2 = Y_c + r \sin (\theta_{m+1} + 90^0) \quad (6.22)$$

For the top face contour start-point $P_3(X_3, Y_3)$ and the top face contour end-point $P_4(X_4, Y_4)$, equations 6.15 to 6.18 can be used.

The cumulative angle of the circular element θ_{m+1} is computed with equation 6.2 and the current point P_2 again will be point P_1 of the next element in the same sequence.

6.3.4 Nature of the Computation Equations

The equations used for computing the coordinates of the contour change-points are derived from the following basic form:

$$X_b = X_a + 1 \cos \theta \quad (6.23)$$

$$Y_b = Y_a + l \sin \theta \tag{6.24}$$

Where $P_a(X_a, Y_a)$ is the reference point while $P_b(X_b, Y_b)$ is the derived point, l is the distance between the two points and θ is the absolute angle of inclination of line P_aP_b with respect to the horizontal axis. The coordinates are absolute as well.

The main advantage of the equations is that they not only yield the correct coordinates, but also impose no restriction upon the range of values they can handle. Thus peculiar cases like zero length and zero radius elements do not constitute any problem, such as overflow during computation, so are cases involving any value of θ.

The basic equations have in fact been derived from the standard parametric equations for circles:

$$X - X_c = R \cos \theta \tag{6.25}$$

$$Y - Y_c = R \sin \theta \tag{6.26}$$

Where $P_c(X_c, Y_c)$ is the centre point, P(X, Y) is a point on the circumference and R is the radius of the circle.

Equations 6.23 and 6.24 may be generalised in the following form:

$$X' = X + \Delta X \qquad (6.27)$$

$$Y' = Y + \Delta Y \qquad (6.28)$$

Where ΔY is a function of ΔX or vice-versa, or perhaps both ΔX and ΔY are functions of some common variable Δt.

The equations may be used for interploation in two-dimensional curves or perhaps also in three-dimensional surfaces if the third dimension variable Z is included.

6.4 Generating the Finished Section Drawing

6.4.1 Layout of the Drawing

For the benefit of the designers, the layout of the drawing has been designed to be similar to the existing ones used in the company. Frames of size from AO to A4 have been used. All the desired information is either printed or drawn on suitable parts of the space enclosed by the frames.

6.4.2 Paper Size and Scale

The size of the available space for drawing depends upon the frame size selected. Therefore scaling facilities have been provided to enable users to control the size of the finished section to be drawn, subject to the limits of the maximum scale permitted with each selected frame size.

6.4.3 The Finished Section Drawing

The finished section drawing is generated according to the data points computed from the definition information supplied for each element. It is automatically drawn to the specified scale in the prelocated space within the frame.

6.4.4 Dimensioning

As the section shape is somewhat unpredictable due to countless possible combinations of linear and circular elements of different sizes and orientations, automatic dimensioning is difficult. Nevertheless, some form of dimensioning facility has been incorporated to aid the designers regarding the identification of each constituent

element of the section. Dimensioning is done on an element by element basis according to the definition sequence type (ORIGIN) selected.

6.5 Element Meanlength and Strip Size

To determine the strip size or width of the raw material from which the ultimate finished section is manufactured, meanlengths of each individual element in the section has to be established. The strip size can then be derived by summing up all the individual meanlengths.

From the engineers' point of view, meanlength of an element refers to the length of that part of the element which remains constant throughout forming.

6.5.1 Meanlength of Linear Elements

Theoretically, linear elements do not undergo deformation in any way during forming since bending does not occur in them. For most practical purposes, such assumption is valid even though it is not strictly true. It is possible for slight deformation to occur in linear elements situated adjacent to circular elements being

bent as a result of the rolling action and occasionally compensating measures may be necessary (see Section 8.2).

6.5.2 Meanlength of Circular Elements

Precise calculation of the meanlength of circular elements is more difficult as both their dimensions and shape may change during forming. The methods used in practice have been based more on empirical experience rather than exact science. Nevertheless, these methods have been claimed to have worked satisfactorily in general.

The most widely used method of deriving the meanlength for a circular element is based on the radius of the element measured from its centre point to the supposed neutral-axis (Fig 6.10), namely the mean radius.

Most experts have suggested that the mean radius (r_m) should be greater than the inside radius (r) by a value ranging from 0.3 of t to 0.5 of t, where t is the material thickness. However, each expert has his own idea of arriving at these values.

According to Walker[17], there are two recommended

methods of obtaining the mean radius (r_m).

The first method is due to Kaltprofile[21], who used the following equation:

$$r_m = r + kt \qquad (6.29)$$

Where r_m is the mean radius, r is the internal radius of bend and k is a factor based on the r/t ratio according to the table below:

r/t	>0.65	>1.0	>1.5	>2.4	>3.8
k	0.30	0.35	0.40	0.45	0.50

TABLE 6.1

MEAN RADIUS FACTOR

The second method is recomended in BS 2994, which is really a variation of the first method:

$$r_m = r + 0.5t_{red} \qquad (6.30)$$

Where

$$t_{red} \equiv \left(\frac{r + kt}{r + 0.5t}\right) t$$

and

$$k = 0.3 \quad \text{for } r/t = 1$$

$$k = 0.35 \quad \text{for } r/t = 1.5$$

The American Society of Mechanical Engineers[7] uses the following formula:

$$BA = (0.0174\ R + 0.0078\ T)\ A \tag{6.31}$$

Where BA = the bend allowance

R = the inside radius

A = the angle of bend in degrees

T = the metal thickness

The formula is effectively equivalent to:

$$r_m = r + 0.4475t \tag{6.31a}$$

if the degree to radian conversion factor and the bend angle are removed.

The American Society for Metals[8] recommends the following:

For $r = 0$,

$$r_m = r + t/3 \quad \text{(normal metal)} \tag{6.32}$$

$$r_m = r + t/2 \quad \text{(less formable metal)} \tag{6.33}$$

For $0 < r < t$,

$$r_m = r + t/3 \tag{6.34}$$

For $r < 2t$,

$$r_m = r + t/2 \tag{6.35}$$

The method of calculation used by the company is similar but is based on different criteria:

$$r_m = r + kt \tag{6.36}$$

Where k is a factor based on the magnitude of the angle of bend (see Table 6.2).

Angle of bend	0 to 80^0	Above 80^0 to 100^0	Above 100^0
k	0.3	0.4	0.5

TABLE 6.2

MEAN RADIUS FACTOR BASED ON ANGLE OF BEND

All the above methods are based upon the assumption that some thinning will occur during bending which is perfectly valid if a constant radius of bend is used for each circular element throughout the bending process, such as in the case of the press-brake process. With Cold Roll-Forming, the assumption holds only if constant radius is also so used. With such kind of bending, thinning is likely to increase with the magnitude of angle of bend and material thickness, which explains why the factor k should be bigger in value to compensate for that, hence equation 6.36 is reasonably valid. In any case, if thinning does occur, the thickness of the bent circular element along its arc is not necessarily uniform, more likely to be thicker at both ends and thinner in the middle. Hence the methods based on the varying thinning criteria cannot be very precisely established.

If on the other hand, bending methods other than the traditional constant radius method can be used to maintain uniform thickness of the section throughout bending, then precise control over the material meanlength can be established. The following analysis illustrates how:

Assuming that control can be exercised during bending such that:

1. Before and after bending, the volume of the material is constant.

2. Longitudinal stretching of the material is negligible.

Then the cross-sectional area (A1) of the finished element shape must be equal to the cross-sectional area (A2) of the starting element shape, as shown in Fig 6.11.

If in addition, uniform thickness t can be maintained throughout forming, then the meanlength of the element (l) can be accurately established as follows:

Consider Fig 6.11 (b),

Let t = element thickness

r = inside radius

θ = angle of bending (in radians)

Then the area of the shaded part (A1):

$$A1 = \frac{\theta}{2\pi} (\pi(r + t)^2 - \pi r^2) = (r + \frac{t}{2}) \theta t \qquad (6.37)$$

Consider Fig 6.11 (a),

Let t = element thickness

l = element meanlength

Then the area of the shaded part (A2):

$$A2 = l t \qquad (6.38)$$

If the material volume is constant, which is normally true, then:

$$A1 = A2$$

Therefore

$$(r + \frac{t}{2}) \theta t = l t$$

or

$$l = (r + \frac{t}{2}) \theta$$

That means the mean radius for t to be uniform or constant, namely without thinning, is:

$$r_m = r + \frac{t}{2} \tag{6.39}$$

The formula applies for any value of r and t.

In theory, the amount of material of each element can be restricted to its legitimate boundaries thereby keeping a constant cross-sectional area throughout forming, to maintain a constant thickness is not impossible. The Opening-Radii technique (see Section 7.3) has been developed in this research solely for this purpose, together with its variations.

6.5.3 Calculation of Strip Size

The strip size S, for m linear elements and n circular elements, may be calculated using the following equation:

$$S = \sum_{i=1}^{m} l_i + \sum_{j=1}^{n} r_j \theta_j \tag{6.40}$$

Where l_i is the meanlength of individual linear elements, r_j is the mean radius of individual circular

elements and θ_j is the final angle of bend (in radians) of the individual circular elements.

6.6 The Program

The software or program for the generation of the finished section drawing consists of two distinctive parts, namely:

1. THE FINISHED SECTION PROGRAM (RPLOT1)

2. THE FRAME MODULE (FRAME)

6.6.1 Program Structure

The hierarchy of the program structure is shown in Chart 6.1, with the main program RPLOT1 residing at the first level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the nature of the subroutines used at each level is given in Table 6.3.

6.6.2 The Frame Module

The module has been specially designed to plot drawing frames conforming to current drawing practices

in the company. The hierarchy of the subroutines in the FRAME module is as shown in Chart 6.2 and a brief account of the nature of each of the subroutines is given in Table 6.4.

All common block data associated with the FRAME module carry labels which start with letters BT.

If new title-block layouts are required, only the FRAME module needs to be modified.

6.6.3 Data Structures

The data structures used for storing the finished section information are as follows:

1. Section Element Definition

   The arrays IDATA (see Table 6.5) and RDATA (see Table 6.6) are used for this purpose.

2. Section Geometry

   The arrays PLOTL and PLOTR (see Table 6.7) are used for top face half-section contour and bottom face half-section contour respectively.

3. Title-Block Content

Table 6.8 shows the array TINPUT for storing title-block content information.

6.6.4 Program Logic Flow

The logic flow of the finished section program (RPLOT1) is summarized in Chart 6.3, using the Symbolic Major Logic Representation (SMLR) notations (see Appendix 2).

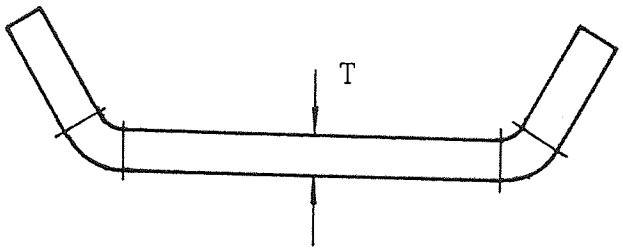


Fig 6.1 ELEMENTS OF THE SECTION GEOMETRY

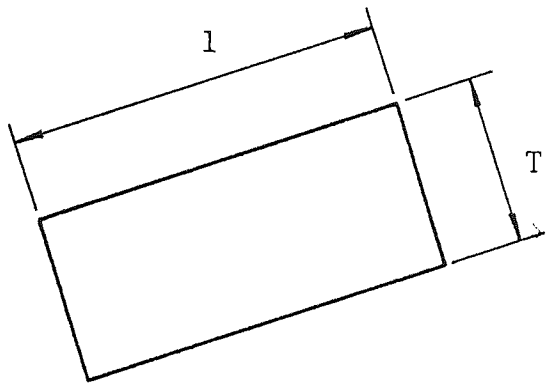


Fig 6.2 DEFINITION OF A LINEAR ELEMENT

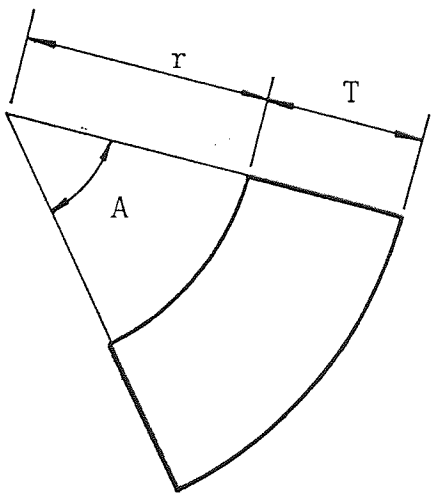


Fig 6.3 DEFINITION OF A CIRCULAR ELEMENT

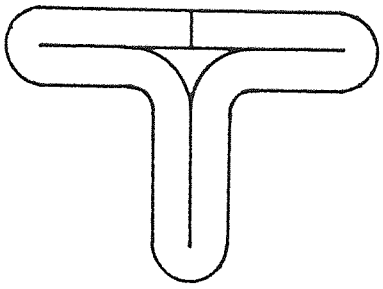


Fig 6.4 A SECTION WITH ZERO RADII ELEMENTS

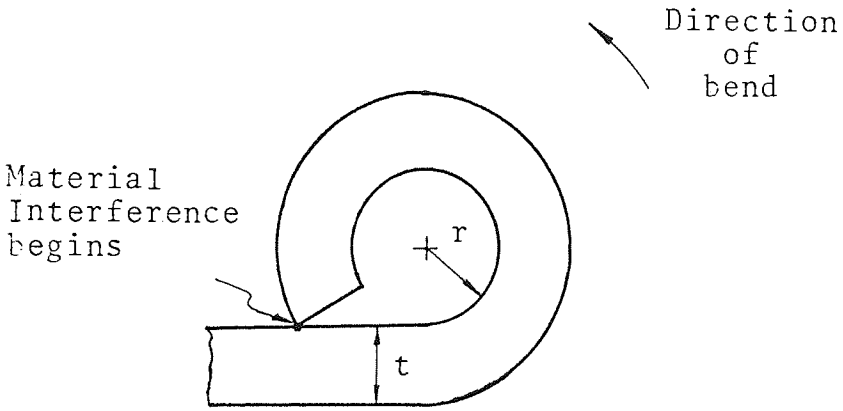


Fig 6.5 MAXIMUM PERMISSIBLE ANGLE OF BENDING FOR CIRCULAR ELEMENTS

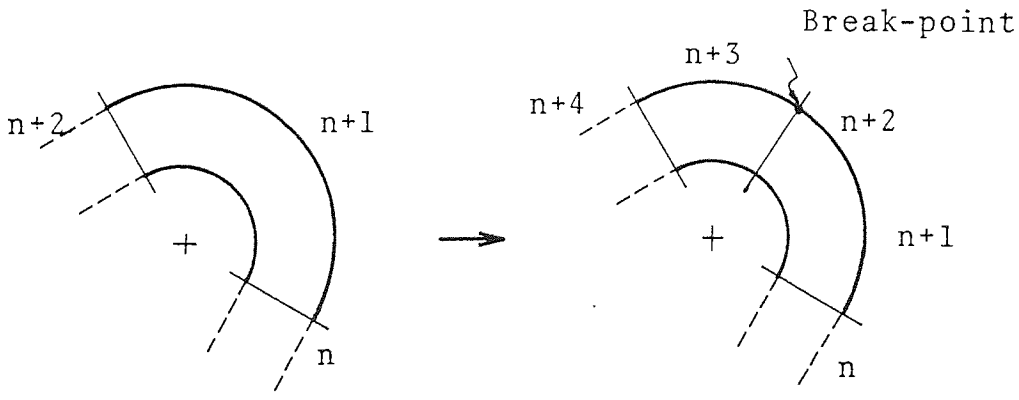


Fig 6.6 COMPOSITE CIRCULAR ELEMENT DEFINITION TO SPECIFY BREAK-POINT

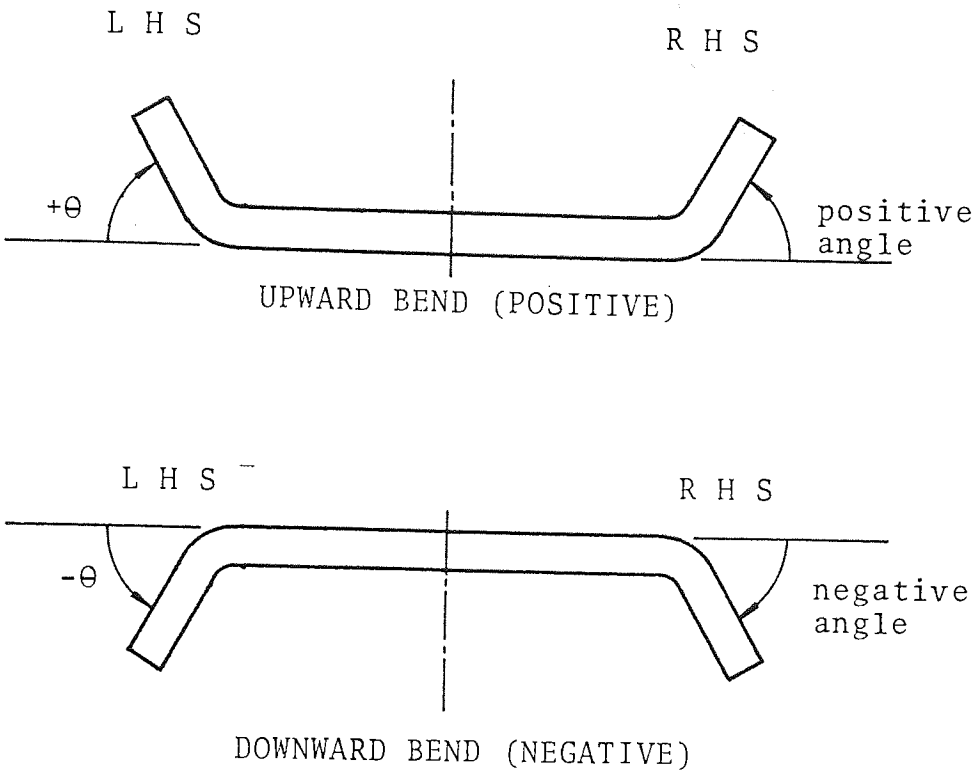


Fig 6.7 CONVENTION FOR THE BENDING DIRECTIONS

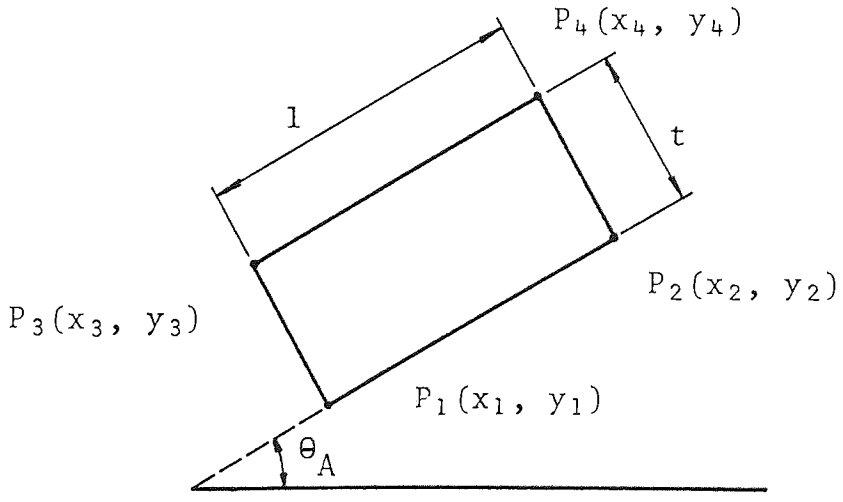
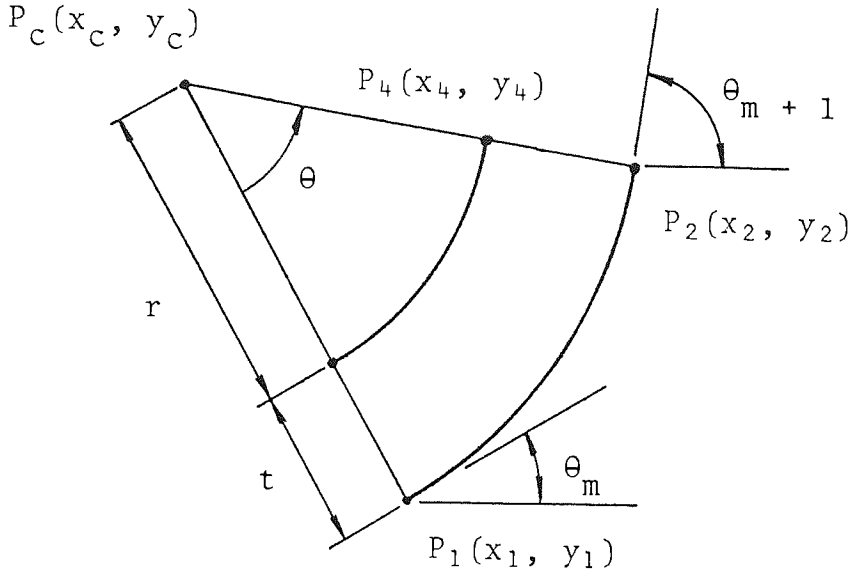
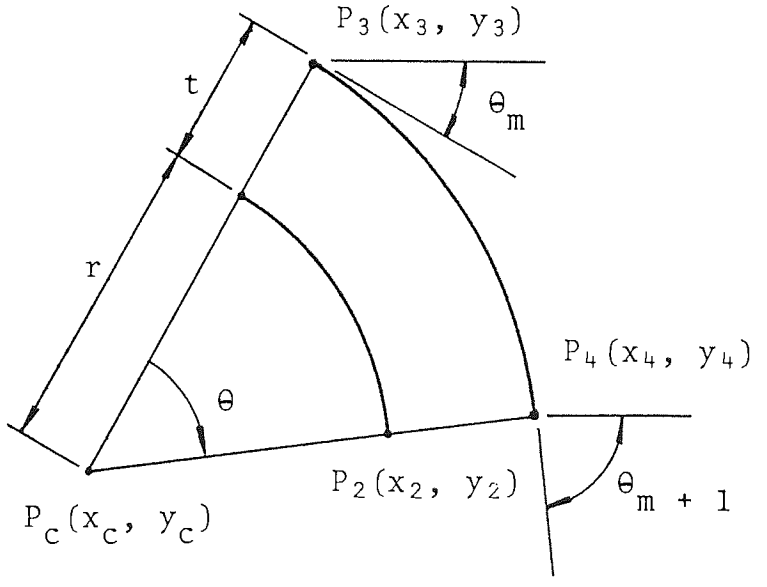


Fig 6.8 COMPUTATION OF LINEAR ELEMENT CONTOUR



(a) UPWARD BEND (θ IS POSITIVE)



(b) DOWNWARD BEND (θ IS NEGATIVE)

Fig 6.9 COMPUTATION OF CIRCULAR ELEMENT CONTOUR

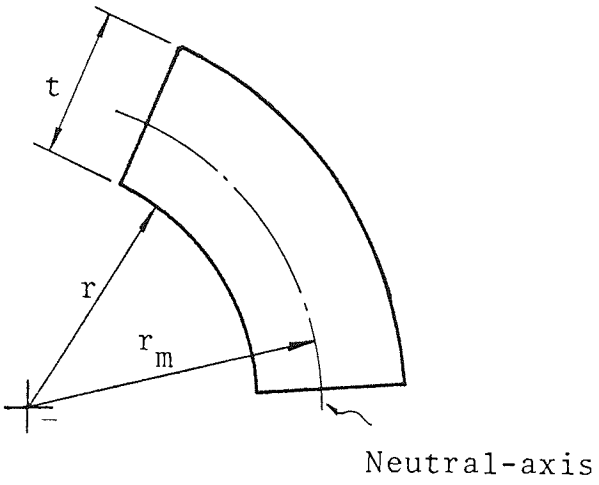
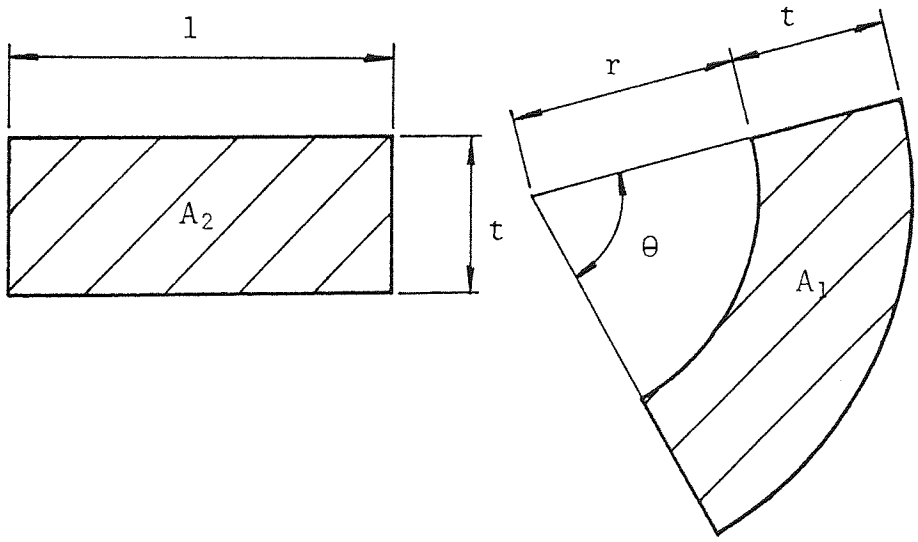


Fig 6.10 MEAN RADIUS OF A CIRCULAR ELEMENT



(a) STARTING SHAPE (b) FINISHED SHAPE

Fig 6.11 MEAN RADIUS BASED ON ELEMENT GEOMETRY

TABLE 6.3 SUBROUTINES OF THE FINISHED SECTION SOFTWARE

NAME	LEVEL	FUNCTION
RPLOT1	1	Calls the subordinate subroutines to decode input data for section definition, to compute the section and draw it with the selected frame size and scale, dimension the drawing if necessary, to compute element meanlengths and material strip size and to output intermediate data into external file for later use.
INIT1	2	Initializes the common block data values.
DCODE	2	Decodes input data for the section element definition sequence.
POSTD	2	Copies the element definition data from first half-section to second half-section (symmetrical section).
MEANL	2	Computes individual element meanlengths and strip size.
FRLM1	2	Computes the space required to draw the finished section.
PLAC1	2	Handles frame size or paper size and scale selection.
PLOT1	2	Draws the finished section.
DTVAL	2	Decode input data for title-block content.
DMEN1	2	Dimensions the drawing.
FRAME	2	(see the FRAME module).
WRTPL	2	Outputs data into intermediate file for later use.

TABLE 6.3 SUBROUTINES OF THE FINISHED SECTION SOFTWARE CONT/D ...

NAME	LEVEL	FUNCTION
STL1	3	Computes the linear element geometry.
ARC1	3	Computes the circular element geometry.
MAXMB	3	Computes the extreme X and Y coordinates of each circular element contour.
DMSTL	3	Dimensions linear elements.
DMARC	3	Dimensions circular elements.
MAXMI	4	Computes the extreme X and Y coordinates of the section.

TABLE 6.4 SUBROUTINES OF THE FRAME MODULE

NAME	LEVEL	FUNCTION
FRAME	1	Calls the subordinate subroutines to draw the various title-blocks and the frame.
TBLK1	2	Draws title-block 1 at the lower left-hand corner.
TBLK2	2	Draws title-block 2 at the lower right-hand corner.
TBLK3	2	Draws title-block 3 at the upper left-hand corner.
TBLK4	2	Prints 'IF IN DOUBT ASK' message.
TBLK5	2	Prints the University's particulars.
PTVAL	2	Calls the subordinate subroutines to print the required title-block content.
PBLK2	3	Prints the content for title-block 2.
PBLK3	3	Prints the content for title-block 3.
CHATL	4	Prints text strings from integer arrays.

TABLE 6.5 THE FIRST SECTION DEFINITION DATA STORE

IDATA(n,2) - INTEGER

SUBSCRIPTS	DESCRIPTION AND VALUES
n,1	Element sequence number, values from 1 to 50.
n,2	Element type, 1 for linear and 2 for circular.

TABLE 6.6 THE SECOND SECTION DEFINITION DATA STORE

RDATA(n,3) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION AND VALUES
n,1	Element length (if linear) or radius (if circular).
n,2	Element bending angle (for circular elements only).
n,3	Cumulative angle of inclination.

(n is the element number forming the first subscript of the arrays, minimum value 1 and maximum value 50).

TABLE 6.7 THE SECTION GEOMETRY DATA STORE

PLOTL(n,8) or PLOTR(n,8) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION AND VALUE
n,1	Start-point X coordinate of circular element contour.
n,2	Start-point Y coordinate of circular element contour.
n,3	Start-point X coordinate of linear element contour.
n,4	Start-point Y coordinate of linear element contour.
n,5	End-point X coordinate.
n,6	End-point Y coordinate.
n,7	Centre-point X coordinate of circular element.
n,8	Centre-point Y coordinate of circular element.

(n is the element number forming the first subscript of the arrays, minimum value 1 and maximum value 50).

TABLE 6.8 THE TITLE-BLOCK CONTENT DATA STORE

TINPUT(m,30) - INTEGER

SUBSCRIPTS	DESCRIPTION AND VALUE
m,1 to m,29	Text for printing under the selected item.
m,30	Number of characters in the text.

(m is the item or selection number as listed in Table 6.9).

TABLE 6.9 SELECTABLE ITEMS IN TITLE-BLOCK CONTENT INPUT

ITEM NUMBER	DESCRIPTION	BLOCK NUMBER	MAXIMUM NUMBER OF CHARACTERS PERMITTED
1	TITLE	2	29
2	CUSTOMER	2	19
3	SEC. NO.	3	14
4	JOB NO.	3	14
5	STRIP SIZE	3	(determined internally)
6	DRAWN	3	9
7	DATE	3	9
8	CHECKED	3	9
9	NO.	4	9
10	MATL.	4	24

CHART 6.1 HIERARCHY OF THE FINISHED SECTION PROGRAM (RPLOT1)

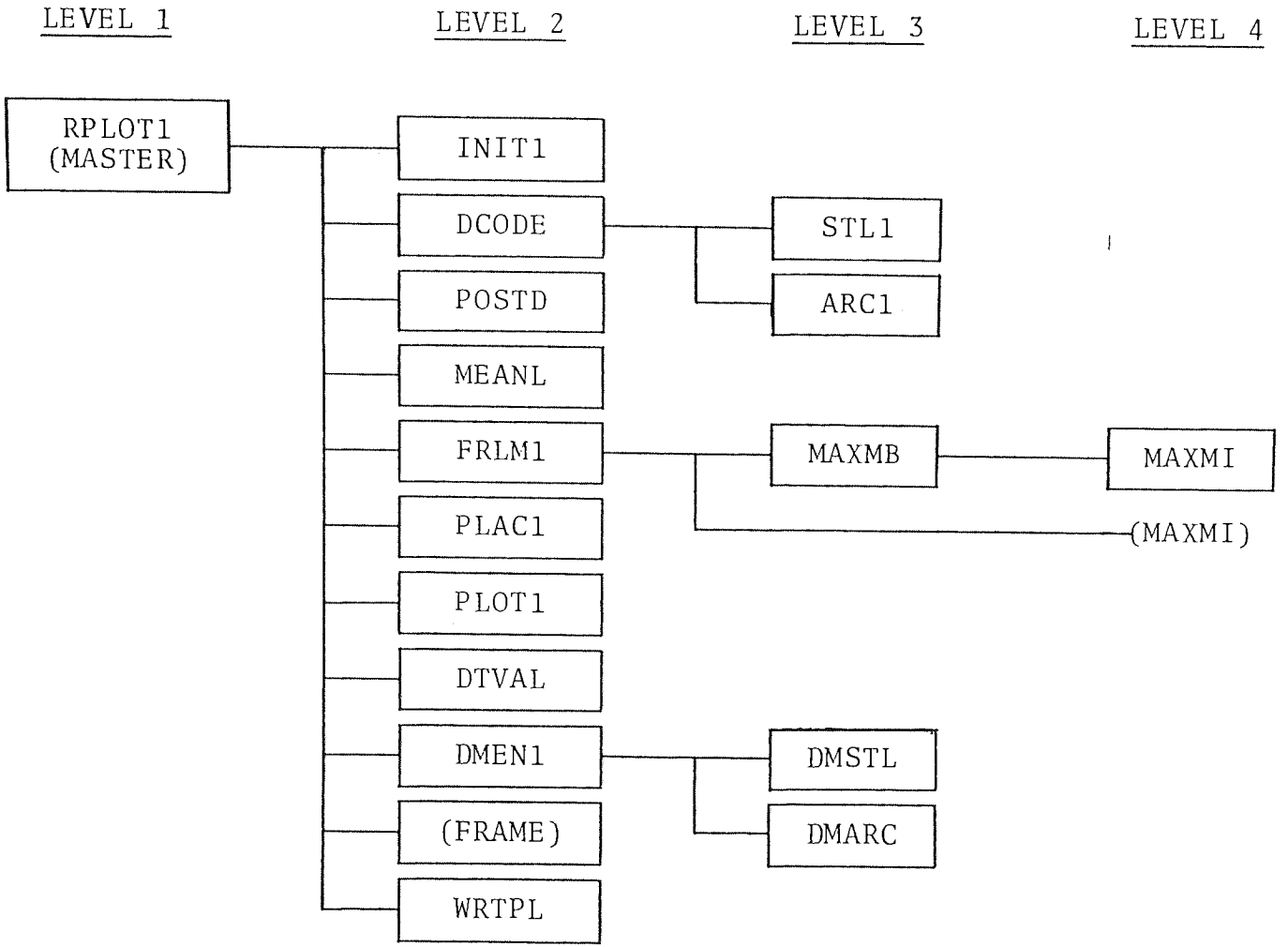


CHART 6.2 HIERARCHY OF THE FRAME MODULE (FRAME)

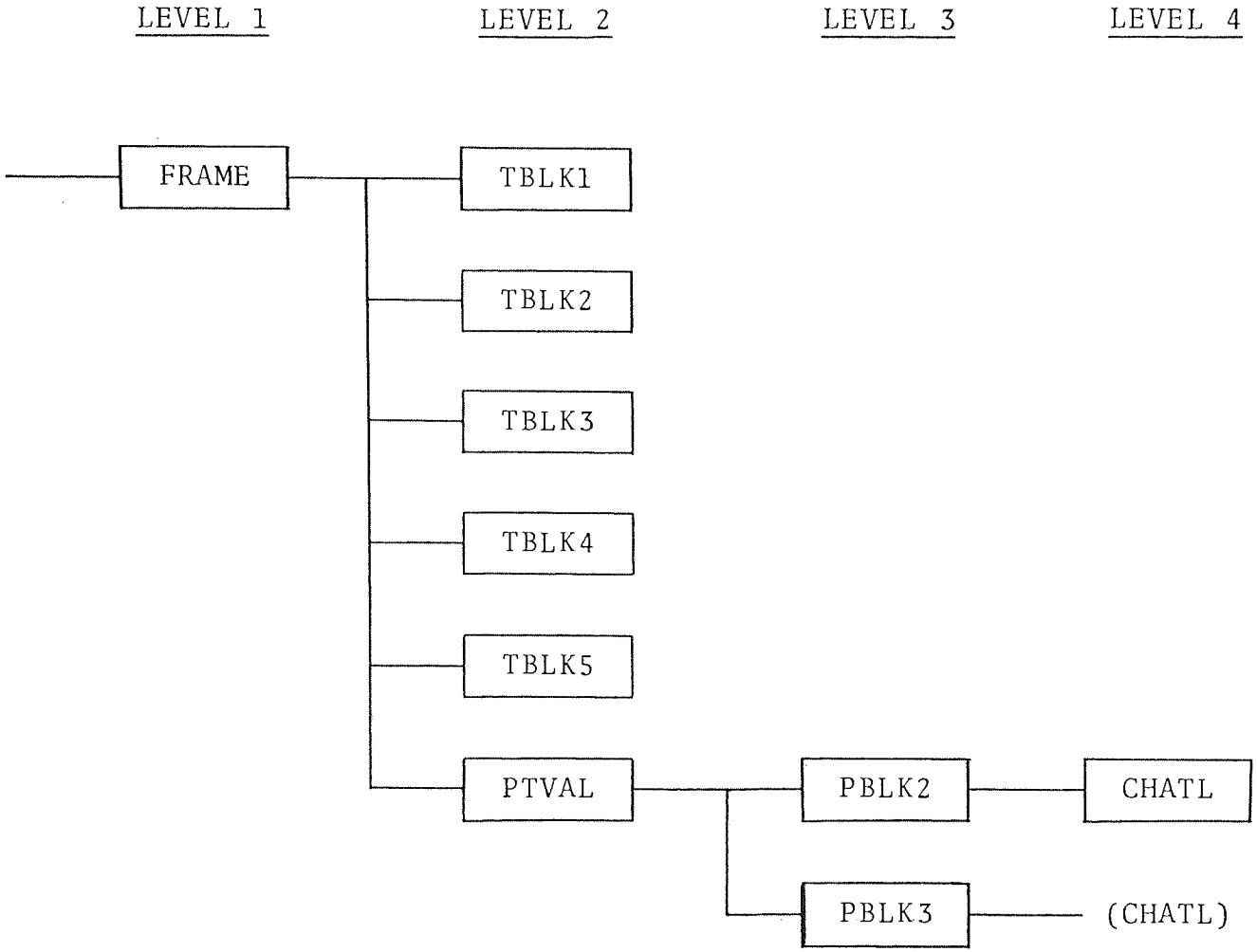


CHART 6.3 LOGIC FLOW OF THE FINISHED SECTION PROGRAM (RPLOT1) (PART 1)

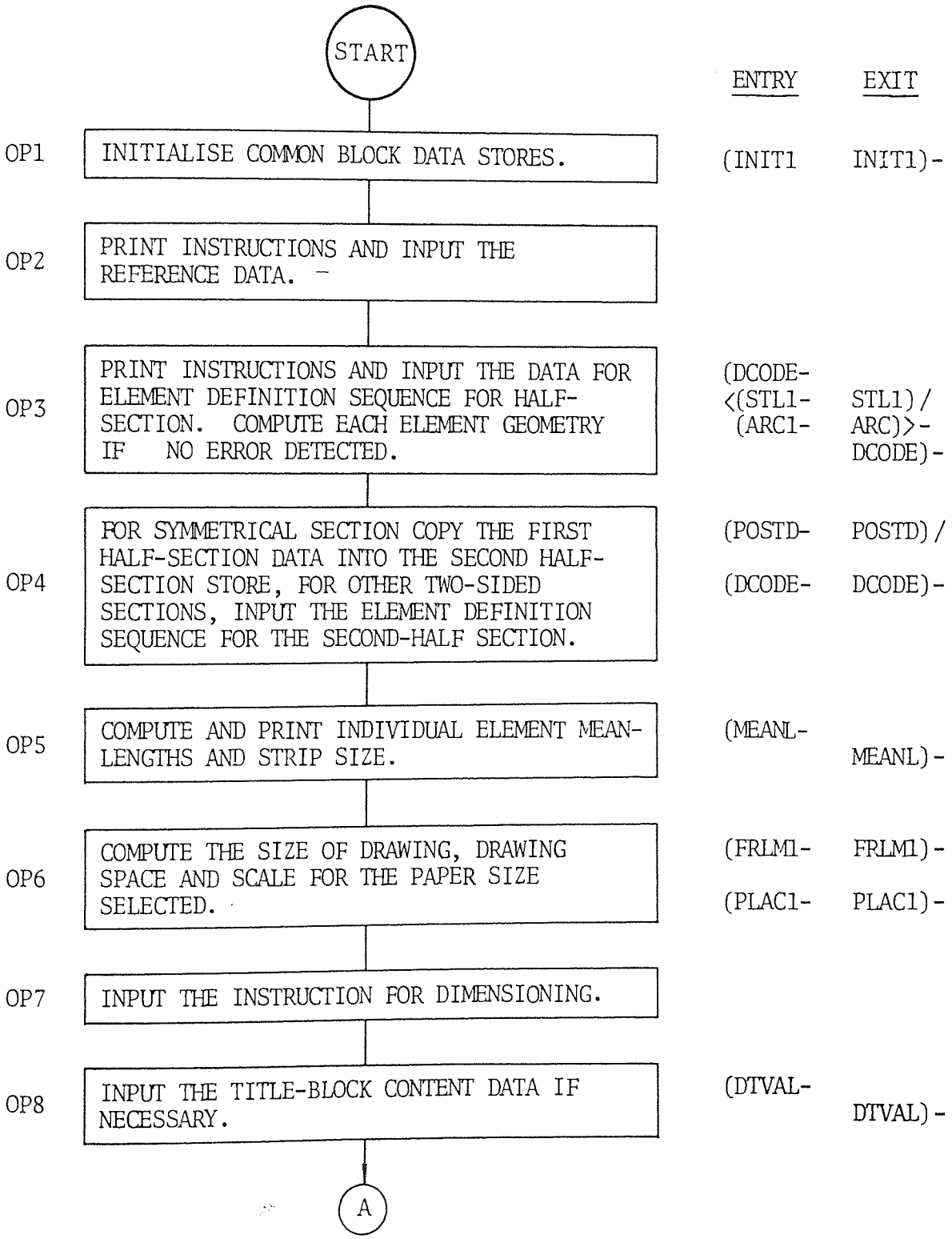
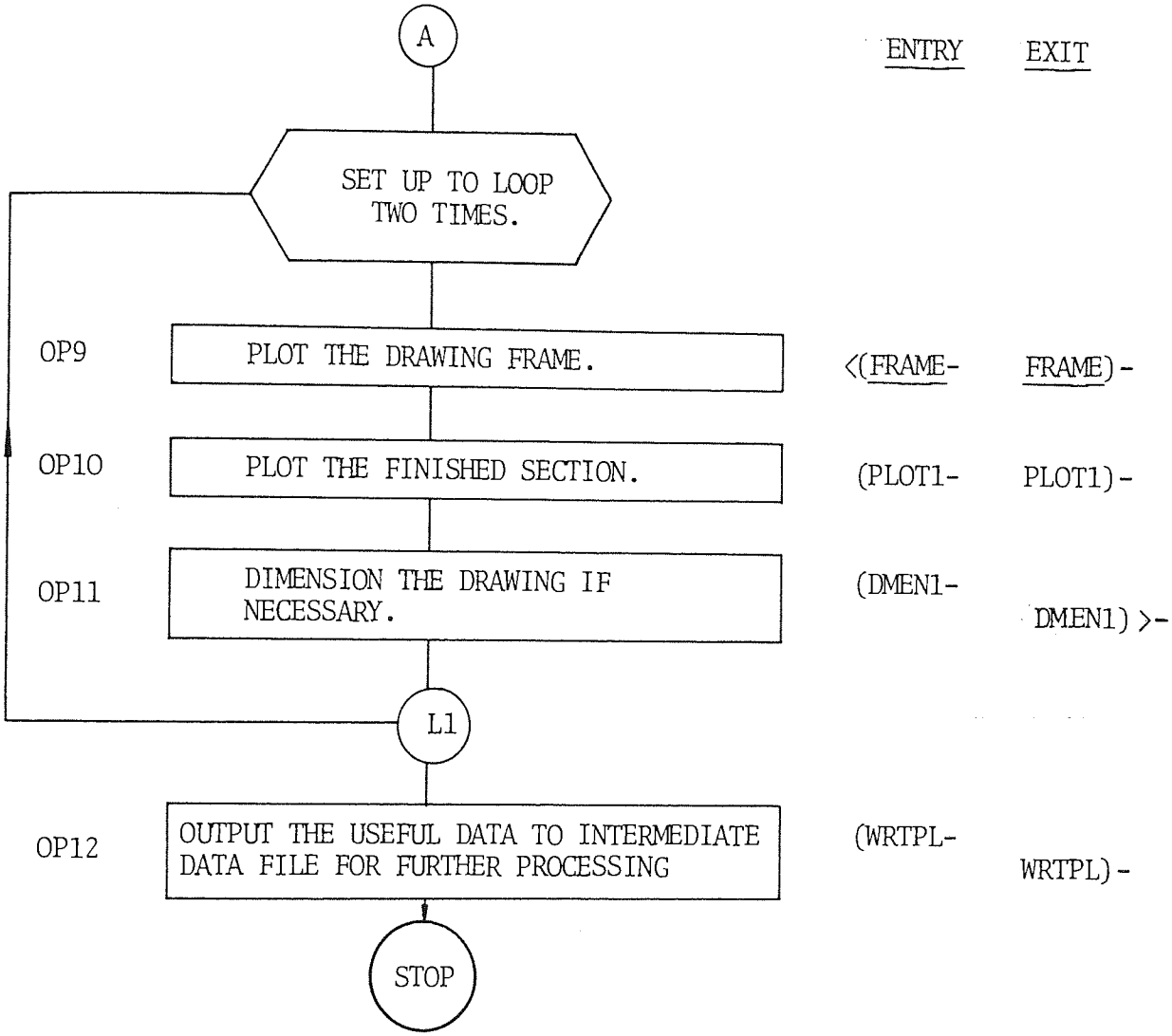


CHART 6.3 (PART 2)



CHAPTER 7

THE FLOWER PATTERN SOFTWARE

7.1 Introduction

Flower patterns are the tools used by roll-designers to produce a satisfactory forming sequence for a particular section. This chapter describes the software developed to automatically generate the flower pattern drawings for roll designers to prove out the forming sequence they design. Element bending methods and the control of bending radii using the Opening-radii method will be discussed in detail.

7.2 Element Bending

The software assumes all elements of the section to be straight at the start of the forming sequence, conforming to the raw material which is a flat strip. During each bending stage or roll-forming stage, certain elements will be bent to certain angular values and should remain so unless they are bent again to new angular values at other subsequent stages. The bending convention described in Section 6.2.5 is used. A system of bending status update for each element has been adopted such that users are required to supply new bending information regarding only

the affected elements, bending status of all other elements in the preceding stage will automatically be assumed in the current stage. Like the section definition scheme described in Section 6.2, the bending definition scheme is also based upon the type of element definition sequence selected (see: Section 6.2.6). For users' convenience, the sides of bending are fixed to be either the left-hand side bending or the right-hand side, rather than the first half-section or the second half-section used in the section definition scheme. Multiple element bending during the same bending stage is permitted, hence as many elements as desired can be bent in one stage. There is also the facility to incline or rotate the section to a non-horizontal position at any bending stage, by specifying the appropriate angular value for element 1.

7.3 Opening-Radii method for monitoring Bending Radius

All the element meanlength values generated in the program RPLOT1 (see: Section 6.6) are retained in an intermediate data file and are used again here for the computation of the flower pattern geometry. As described in Section 6.5.2, meanlength of all circular elements can be computed very precisely using the constant material volume principle. In that way, constant cross-sectional

area is maintained for each element throughout all the bending stages. It has also been illustrated that to maintain a uniform section thickness, equation 6.39 applies:-

$$r_m = r + \frac{t}{2} \qquad (6.39)$$

where r_m is the mean radius, r is the inside radius and t is the thickness.

The corresponding meanlength is therefore:-

$$L = (r + \frac{t}{2})\ \theta \qquad (7.1)$$

To maintain a constant cross-sectional area and uniform material thickness throughout forming, L can be kept constant at all bending stages. The inside radius of bend r can then be varied or derived according to the value of θ at each bending stage:-

$$r_i = \frac{L}{\theta_i} - \frac{t}{2} \qquad (7.2)$$

where r_i is the inside radius of bend at stage i and θ_i is tha angle of bend (in radians) at stage i.

Provided that the relative positions of the elements adjacent to the particular circular element being bent can

be precisely monitored at each bending stage, it is theoretically feasible to maintain a constant cross-sectional area and a uniform thickness for the circular element throughout forming, as illustrated in Fig 7.1, such that:-

$$r_i\theta_i = r_j\theta_j = \ldots\ldots k \qquad (7.3)$$

where k is constant.

This method of controlling the radii of bend during forming has come to be called the OPENING-RADII method, previously not used by the company because it is a longer method than the method of keeping the radius of bend of each element always constant. However, after its introduction as a computerised method in this research, satisfactory results have been accomplished consistently by the users in most cases, therefore its validity is confidently confirmed. The only situation where the Opening-Radii method cannot be applied satisfactorily is the bending of very short legs, legs of length typically less than five times the section thickness, because the larger radii of bend generated by the method for the beginning bending stages cannot effectively control the slipping and springback of the material. To overcome that, a second method of bending using the composite

circular element length definition to yield smaller radii of bend has since been introduced (see: Section 8.3).

The main advantages of using the Opening-Radii method are that it enhances the wear characteristics of the roll due to the use of larger radii of bend and that it permits smoother forming of circular elements thus reducing the tendency of thinning.

7.4 Flower Patterns

Two types of flower patterns are produced by the software. The first type is a flower pattern with a common origin (see: Fig 4.5) while the second type is a flower pattern with separate origins (see: Fig 4.6). With the flower patterns, roll designers can project each of their forming sequence designs in drawing and perform visual inspection and alterations until their satisfaction. The flower patterns are drawn to be proportionally accurate with automatic scaling so that interference of material as a result of certain bends as well as awkward bends which prevent roll contact can be easily detected and corrected. At present much of the design decisions still rests upon the designers themselves regarding the design of satisfactory forming sequences but with accurate flower

patterns as faithful representation of the designs, quicker and more accurate judgement can be exercised.

7.5 Generating the Flower Pattern Drawings

The flower pattern shapes are generated one by one in accordance with the sequence of forming, starting with a flat strip. The meanlength of each element is obtained from the intermediate data file via input and normally remain unchanged throughout forming except in situations where length adjustments become necessary (see: Section 8.2). All elements start with zero angle of bending, namely they are all flat. Based on the bending definition input data associated with each successive bending stage, the angular values of the bent elements are updated accordingly when specifically required. Unless updated, the angular value of each bent element will remain unaltered, conforming to the actual forming outcomes in practice. The computation procedure for the flower pattern geometry is also based upon equations 6.23 and 6.24 used for computing the finished section geometry and is described as follows.

7.5.1 Linear Element Computation

In Fig 7.2, the known quantities are $P_1(X_1, Y_1)$, the

end point of the previous element, and θ_m, the cumulative angle of inclination at P_1. The element end-point $P_2(X_2, Y_2)$ can be derived using the following equations:-

$$X_2 = X_1 + 1 \cos \theta_m \quad (7.4)$$

$$Y_2 = Y_1 + 1 \sin \theta_m \quad (7.5)$$

where 1 is the element meanlength.

7.5.2 Circular Element Computation

In Fig 7.3 (a) and (b), the known quantities are again $P_1(X_1, Y_1)$, the end-point of the previous element, and θ_m, the cumulative angle of inclination at P_1. Knowing both the element meanlength L and the angle of bend θ, the radius r_m may then be calculated:-

$$r_m = \frac{L}{\theta} \quad (7.6)$$

Then the centre point $P_c(X_c, Y_c)$ may be derived:-

(1) For $\theta > 0$,

$$X_c = X_1 + r_m \cos (\theta_m + 90^0) \quad (7.7)$$

$$Y_c = Y_1 + r_m \sin (\theta_m + 90^0) \quad (7.8)$$

(2) For $\theta < 0$,

$$X_c = X_1 + r_m \cos (\theta_m - 90^0) \quad (7.9)$$

$$Y_c = Y_1 + r_m \sin (\theta_m - 90^0) \quad (7.10)$$

Then end-point $P_2(X_2, Y_2)$ may also be derived:-

(1) For $\theta > 0$,

$$X_2 = X_c + r_m \cos (\theta_m + \theta - 90^0) \quad (7.11)$$

$$Y_2 = Y_c + r_m \sin (\theta_m + \theta - 90^0) \quad (7.12)$$

(2) For $\theta < 0$,

$$X_2 = X_c + r_m \cos (\theta_m + \theta + 90^0) \quad (7.13)$$

$$Y_2 = Y_c + r_m \sin (\theta_m + \theta + 90^0) \quad (7.14)$$

7.6 The Program

The software or program for the generation of the flower pattern drawings consists of four distinctive parts, namely:-

(1) THE FLOWER PATTERN PROGRAM (RPLOT2)

(2) THE FRAME MODULE (FRAME)

(3) THE FLOWER PATTERN INPUT DECODER (DCOFR)

(4) THE ELEMENT LENGTH MONITOR (CLENG)

The FRAME module has been described in Section 6.6.2. The Flower Pattern Input Decoder will be described in Section 7.6.2 and the Element Length Monitor will be described in Section 8.7.2.

7.6.1 Program Structure

The hierarchy of the program structure is shown in Chart 7.1, with the main program RPLOT2 residing at the first level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the nature of the subroutines is given in Table 7.1.

7.6.2 The Flower Pattern Input Decoder

This module is designed to deal with all input data from the Flower Pattern Input Data File (FXXXX). The hierarchy of the subroutines in this module is as shown in Chart 7.2 and a brief account of the nature of each of the subroutines used is given in Table 7.2.

7.6.3 Data Structure

The data structure peculiar to the flower pattern software is in the form of arrays which store the information about the geometry of the flower pattern at a certain bending stage on an element by element basis. The arrays are organised into left-hand and right-hand half-sections. Table 7.3 describes the content of the arrays.

7.6.4 Program Logic Flow

The logic flow of the flower pattern program (RPLOT2) is summarized in Chart 7.3, using the Symbolic Major Logic Representation (SMLR) notations (see: Appendix 2).

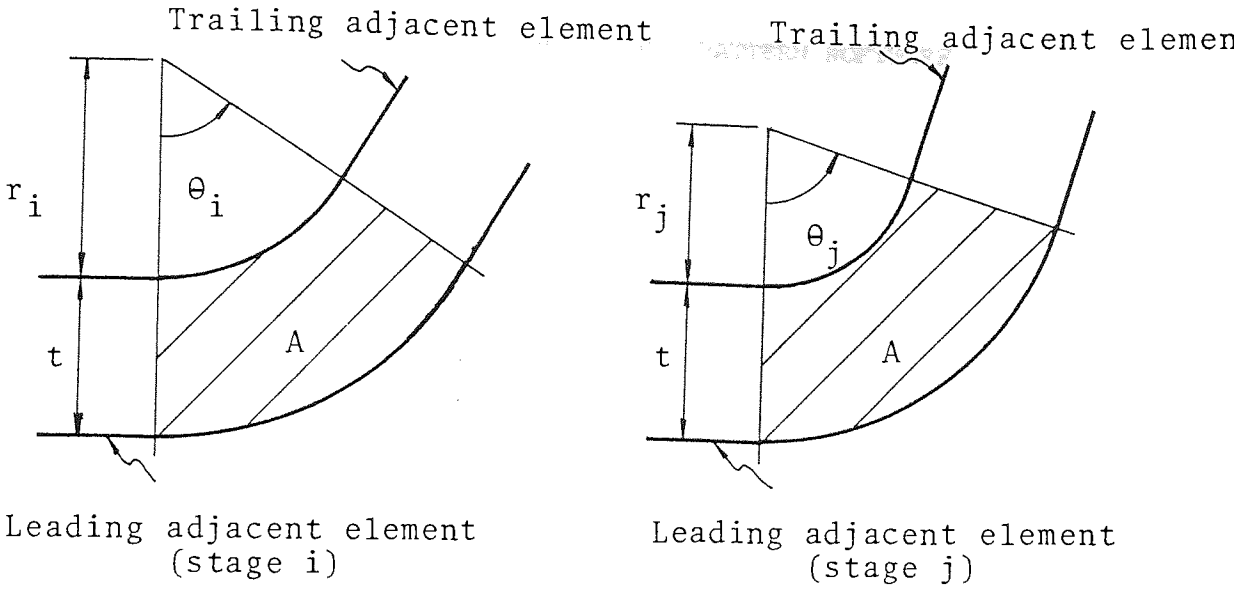


Fig 7.1 THE OPENING-RADII METHOD OF FORMING

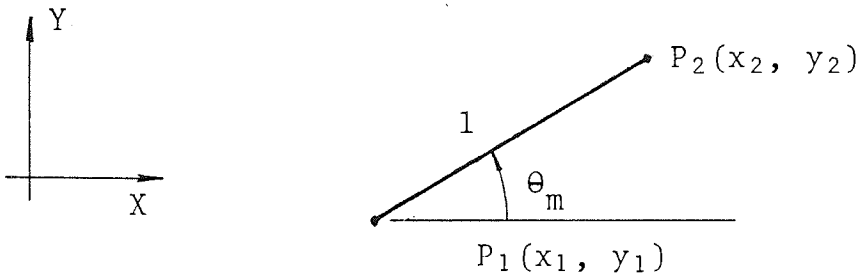
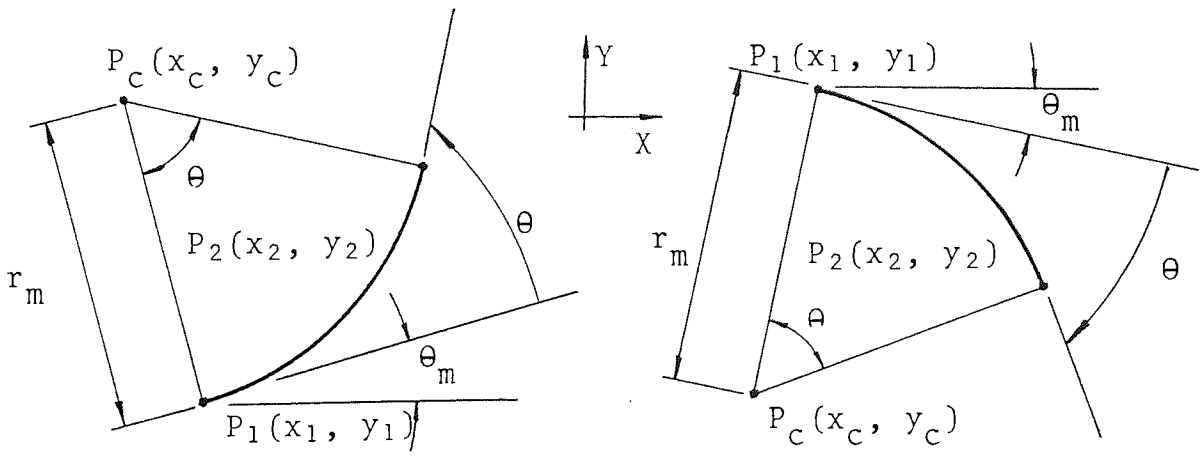


Fig 7.2 LINEAR ELEMENT OF THE FLOWER PATTERN



(a) θ is positive (b) θ is negative

Fig 7.3 CIRCULAR ELEMENT OF THE FLOWER PATTERN

TABLE 7.1 SUBROUTINES OF THE FLOWER PATTERN SOFTWARE

NAME	LEVEL	FUNCTION
RPLOT2	1	Calls the subordinate subroutines to decode input data for defining element bending and element length monitoring at each bending stage and to automatically draw the two types of flower pattern in separate frames.
RDPL	2	This subroutine inputs reference data from the intermediate data file.
DCOFR	2	(see the Flower Pattern Input Decoder)
FLOWR	2	Control subroutine for managing flower pattern generation. The flower patterns are computed twice. In the first round the scale for each flower pattern is established. In the second round each shape is plotted in the respective frames.
COMPF	3	Computes one flower pattern geometry for one particular bending stage.
FRLM2	3	Establishes the largest X and Y dimensions of each flower pattern shape.
PLAC2	3	Allocates the space and determines the scale for drawing flower pattern A.
PLAC3	3	Allocates the space and determines the scale for drawing flower pattern B.
FRAM1	3	Draws the frames and prints the headings.
PLOT2	3	Draws each individual shape of the flower patterns.
STGUP	4	Updates the bending status of the specified elements to new angular values for the current bending stage.

TABLE 7.1 SUBROUTINES OF THE FLOWER PATTERN SOFTWARE CONT/D ...

CLENG	4	(see the Element Length Monitor).
FWRPT	4	Computes the flower pattern geometry.
MAXMA	4	Computes the actual space occupied by circular elements.
FRAME	4	(see the FRAME module).
MAXMI	5	(see Table 6.3).

TABLE 7.2 SUBROUTINES OF THE FLOWER PATTERN INPUT DECODER

NAME	LEVEL	FUNCTION
DCOFR	1	Decodes some of the input data and controls the decoding format according to the selected definition options.
BENDT	2	Decodes bending status data without composite element length definition.
BENDX	2	Decodes bending status data with composite element length definition.
BENDY	2	Decodes instructions for the radii-sharpening option.
BENDZ	2	Decodes instructions for the short-leg bending option.

TABLE 7.3 THE FLOWER PATTERN DATA STORE

PFWRX(n,8) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION AND VALUE
n,1	Current angle of bending (circular elements).
n,2	Current cumulative angle of bending.
n,3	X-coordinate of centre-point (circular elements).
n,4	Y-coordinate of centre-point (circular elements).
n,5	X-coordinate of end-point.
n,6	Y-coordinate of end-point.
n,7	Direction of arc (circular elements).
n,8	Spare.

(n is the element number forming the first subscript of the arrays, minimum value 1 and maximum value 50).

CHART 7.1 HIERARCHY OF THE FLOWER PATTERN PROGRAM (RPLOT2)

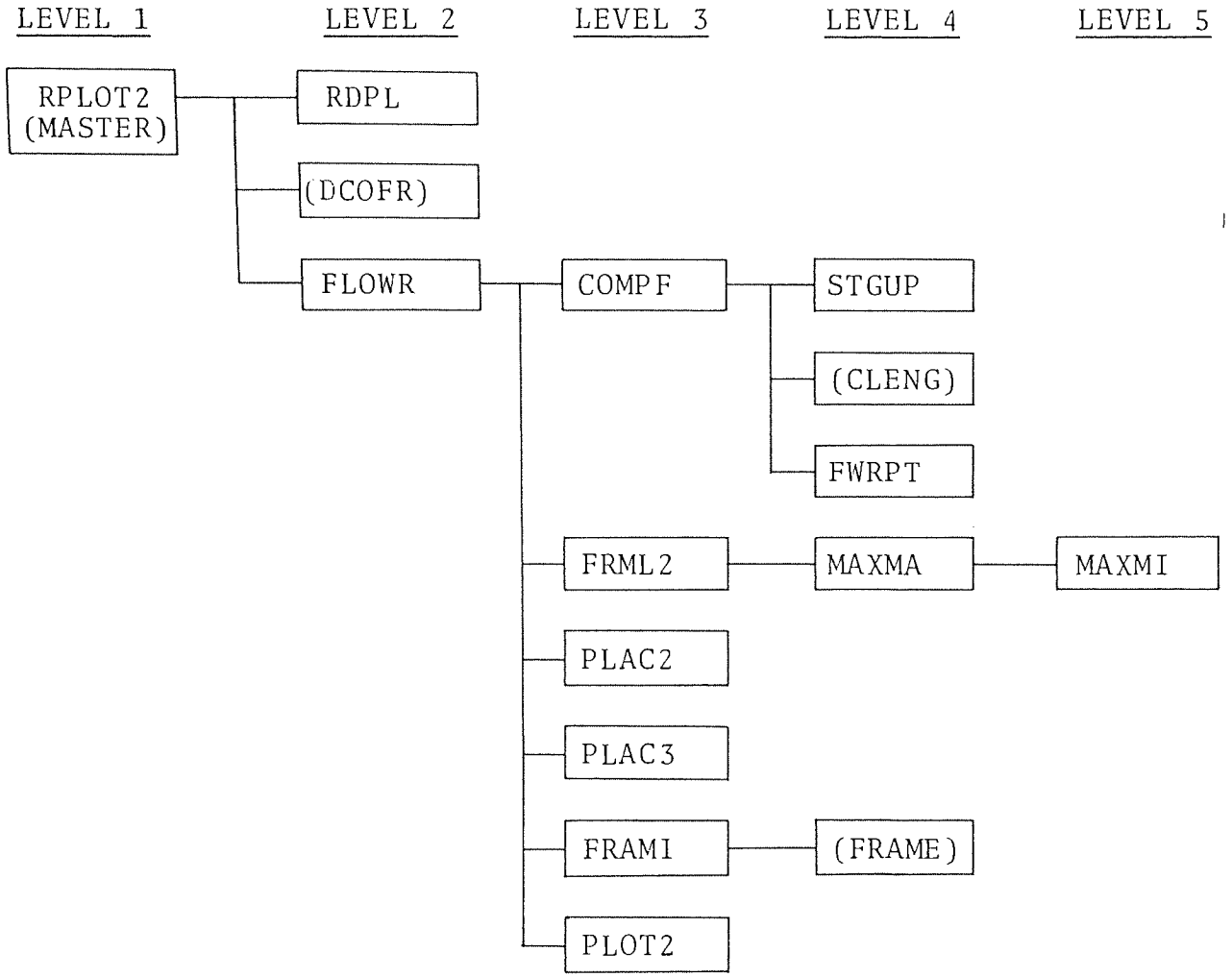


CHART 7.2 HIERARCHY OF THE FLOWER PATTERN INPUT DECODER (DCOFR)

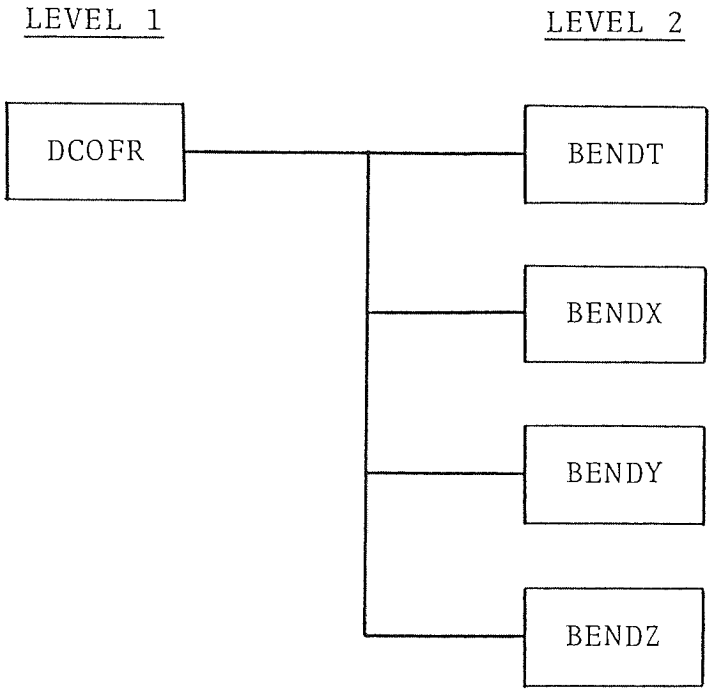


CHART 7.3 LOGIC FLOW OF THE FLOWER PATTERN PROGRAM (RPLOT2) (PART 1)

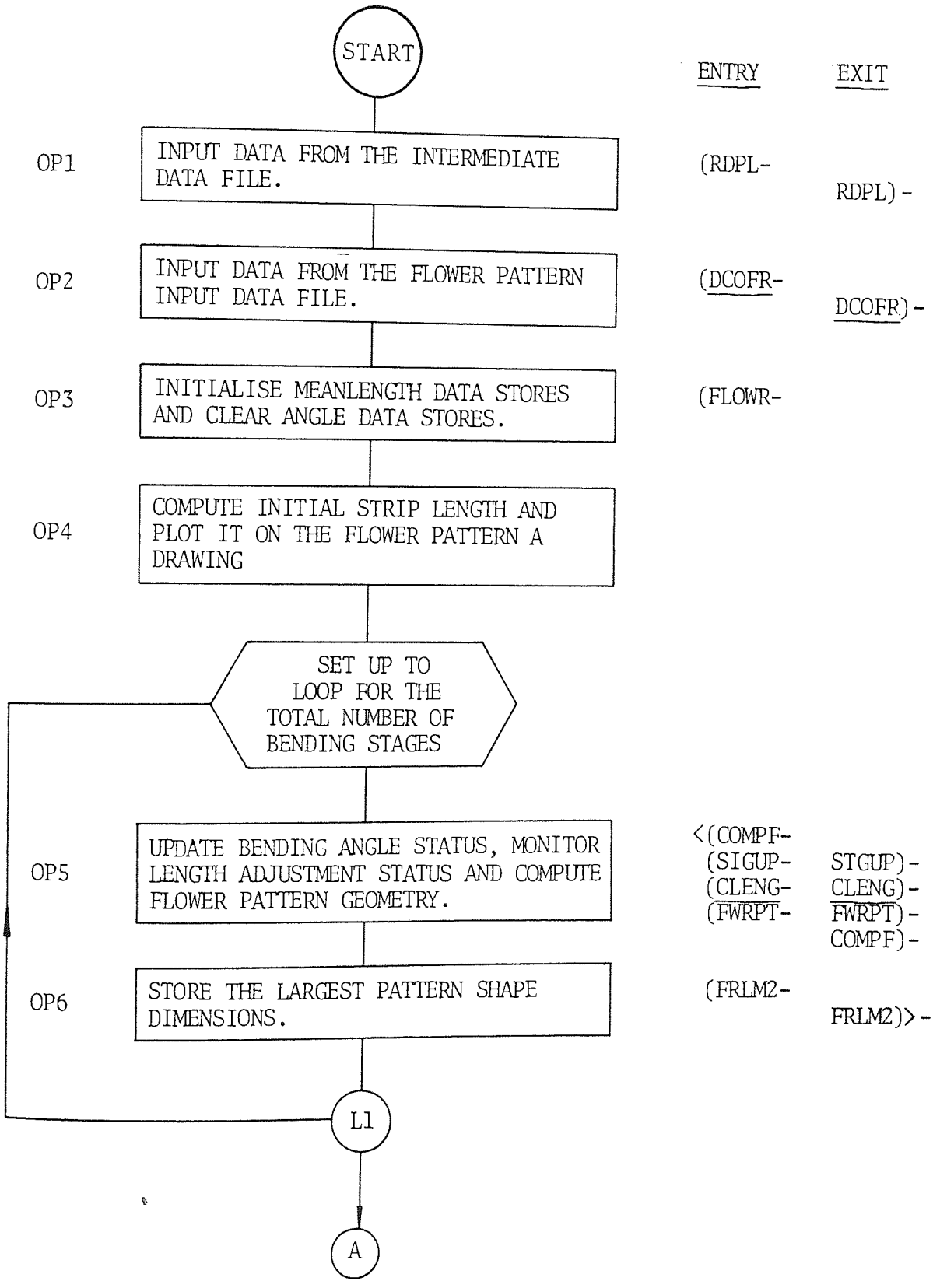
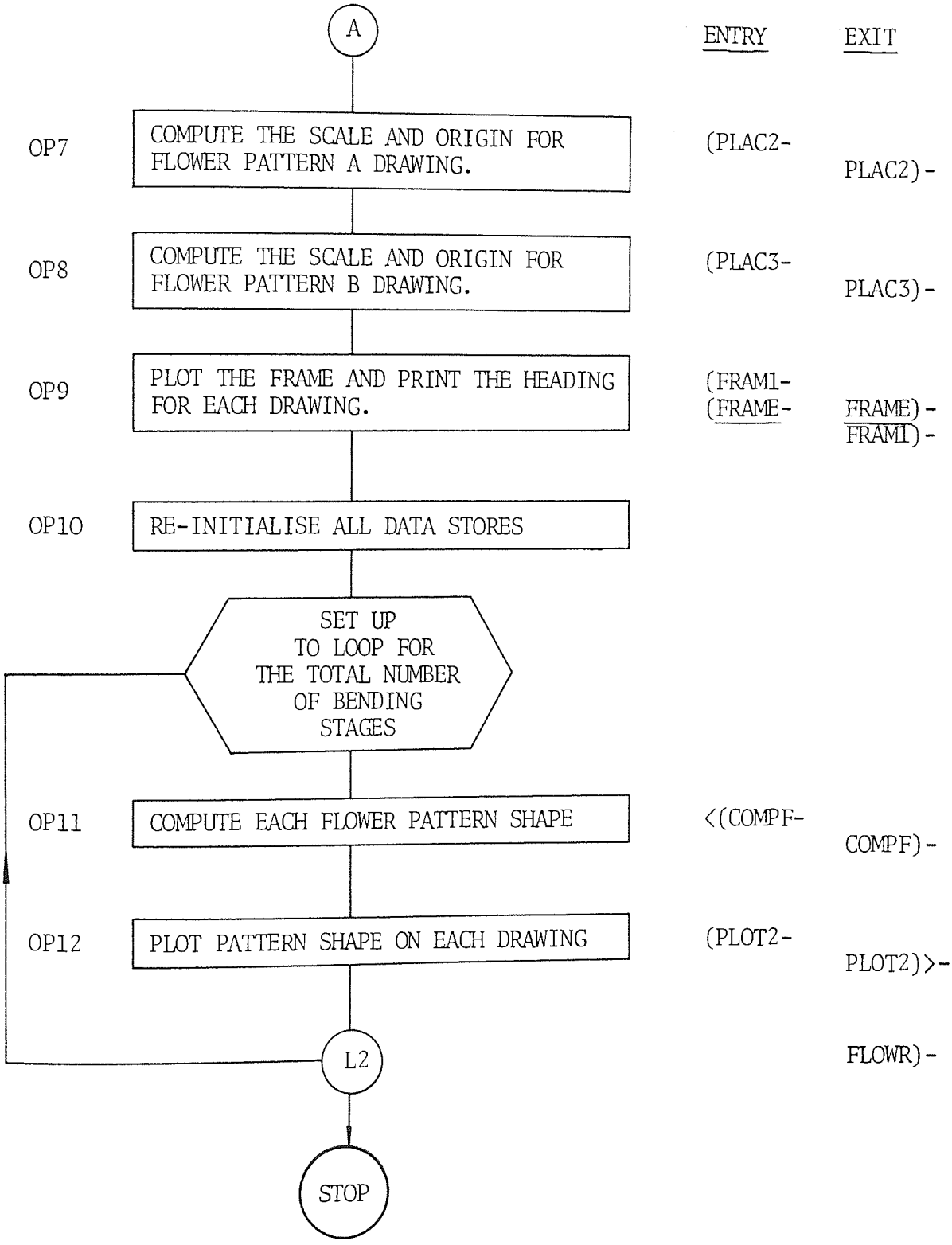


CHART 7.3 (PART 2)



CHAPTER 8

THE 10 TO 1 TEMPLATE SOFTWARE

8.1 Introduction

As a conventional method, template contour design is a vital and integral part of roll design and manufacture. It is therefore logical to provide whatever aids the designers require in producing satisfactory and acceptable template designs in the form of 10 to 1 drawings (see Fig 4.7) while the use of wire templates is still indispensible.

For forming requirements, occasionally the exact shape and size of particular elements being formed have to be modified for more satisfactory forming performance. One of the frequently adjusted parameters is element length, hence facilities to allow designers to do that are considered essential. At times, smaller radii may be required at the bent element, rather than the larger radii produced by the Opening-Radii method (see Section 7.3). To maintain the uniform thickness characteristic throughout forming, part of the bent element at intermediate forming stages has to remain linear if smaller radius is used, the element effectively

becomes a composite element with some part linear and some part circular. A scheme for designing templates using composite length definition has therefore been included for this purpose. In addition, facility to slightly sharpen the inside radii of the bent elements has also been included for improving the wear characteristics of the rolls produced from the templates. All these features and the software which handles them will be fully described in this chapter.

8.2 Element Length Adjustment

Depending on the way an element is bent, the behaviour of the flow of the strip material during forming varies. It is normal practice to bend an element upwards or downwards from a straight shape in horizontal orientation to about 45^0 with only the top and bottom rolls, such action is asserted to cause material from the stationary end of the bent element to move into the bent region (Fig 8.1). To bend an element beyond 45^0, normally side-rolls are used and that is asserted to have the opposite effect of causing the material from the bent region to move to the stationary end (Fig 8.2).

During roll-forming, it is usually not possible to exert sufficient clamping pressure on the linear parts A and B (Figs 8.1 and 8.2) to stop the material movement to or from the bent region, hence adjustment of the element length becomes necessary. In practice, part A is made slightly longer for initial bends of low angular values using top and bottom rolls, to compensate for the material dragged into the bent region. As the angles of bend is large enough for the use of side-rolls, part A is then shortened to cancel the amount of material dragged out of the bent region.

8.3 Composite Element Length Definition Option

For circular elements, there are times when smaller radius is more desirable than larger radius with the same angle of bend, mainly for positively restricting undesirable and excessive movement of material during forming. By making the radius of a circular element smaller, while both the angle of bend and the thickness remain constant, part of the element will have to be excluded from bending and remain unbent or linear, normally at both ends of the element. Effectively the circular element becomes a three-part composite element (Fig 8.3). With only a few exceptions, such kind of bend radius

adjustment while keeping the element meanlength constant can usually be applied. The exceptions are cases when as a result of adjustment the radius may become negative because the initial radius is small. One particular application where adjustment to sharper radius is necessary is the bending of very short legs, typically of length less than five times the thickness, for overcoming slipping and springback of the material.

The computation for the composite element involves only the distribution of the meanlength of the original circular element to three constituent parts P_1, P_2 and P_3 (Fig 8.3 (b)). P_1 is then added to the leading adjacent element X and P_3 is then added to the trailing adjacent element Y, while P_2 still remains circular. All these parts are specified with percentage values defined by input data to the software and their values are updated stage by stage. With such an arrangement, the computation procedure used for the individual element geometry originally can still be applied without any modification.

8.4 Radii-Sharpening Option

The purpose of radii-sharpening is two fold. Firstly it increases the area of high wear at the convex part of

the roll contour, namely the inside radii of the bent elements, thereby improving the wear characteristics. Secondly, it can be used with the control of roll pressure against the strip to monitor the clearance between the roll surface and the strip surface for better pinch-difference (see Section 9.3) control. The radii-sharpening operation involves redistributing the mean-length of the existing circular part of a simple or composite element to three further constituent parts rather similar to the composite element length procedure. In that way both the composite element length operation and the radii-sharpening operation may co-exist without interference on one another. The software has been designed such that users may switch the radii-sharpening operation on or off at any bending stage, except the last bending stage when it is always off since that is when the exact section geometry is required.

The computation for the radii-sharpening operation is as follows:

In Fig 8.4, the length of the linear part l_1 is fixed according to a predetermined percentage P of the original arc length.

Knowing the original arc length c_1 and the angle of bend θ, the original mean radius R_1 may be derived:

$$R_1 = c_1/\theta \tag{8.1}$$

Also in Fig 8.4,

$$l_3 = R_1 \tan \frac{\theta}{2} \tag{8.2}$$

And by similar triangles,

$$\frac{R_2}{R_1} = \frac{l_3 - l_1}{l_3} \tag{8.3}$$

The arc length c_2 of the new circular part with the smaller radius R_2 can then be established:

$$c_2 = R_2 \theta \tag{8.4}$$

At the company's request, the radii-sharpening percentage has been fixed at 30% reduction of the original radii values.

8.5 Short-Leg Bending Option

This has been intended as a short-hand way of defining

the composite element lengths using fixed percentages of the linear parts for the required number of bending stages. The percentages in this case are 50% for the linear parts and 50% for the circular part of the specified element or elements. The scheme is necessary for bending short-legs of length typically less than five times the material thickness.

8.6 Generating the 10 to 1 Template Drawings

The method used in generating the 10 to 1 template drawings is actually a combination of the method used for the flower patterns and the method used for the finished section. Effectively the procedure for computing the section geometry is repeatedly executed for different angles of bend at different bending stages according to the flower pattern bending definitions. Each of the intermediate section so generated is drawn ten times full size provided the paper width limit is not exceeded, otherwise the drawing is scaled down automatically.

8.7 The Program

The software or program for the generation of the 10 to 1 template drawings consists of four distinctive

parts, namely:

1. THE 10 TO 1 TEMPLATE SOFTWARE (RPLOT3)

2. THE FLOWER PATTERN INPUT DECODER (DCOFR)

3. THE ELEMENT LENGTH MONITOR (CLENG)

The Flower Pattern Input Decoder has been described in section 7.6.2, while the Element Length Monitor will be described in section 8.7.2.

8.7.1 Program Structure

The hierarchy of the program structure is shown in Chart 8.1, with the main program RPLOT3 residing at the first level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the subroutines used is given in Table 8.1.

8.7.2 The Element Length Monitor

This is the module that monitors the length of all the section elements at each bending stage according to the instructions given by users via input for composite element length definition, radii-sharpening and short-leg bending. The hierarchy of the subroutines in this

module is as shown in Chart 8.2 and a brief account of the subroutines involved is given in Table 8.2.

8.7.3 Data Structure

The data structure used for the template contour data is identical to that of the finished section (see Table 6.7), so is the data structure used for the element definition data.

8.7.4 Program Logic Flow

The logic flow of the 10 to 1 template program (RPLOT3) is summarized in Chart 8.3, using the Symbolic Major Logic Respresentation (SMLR) notations (see Appendix 2).

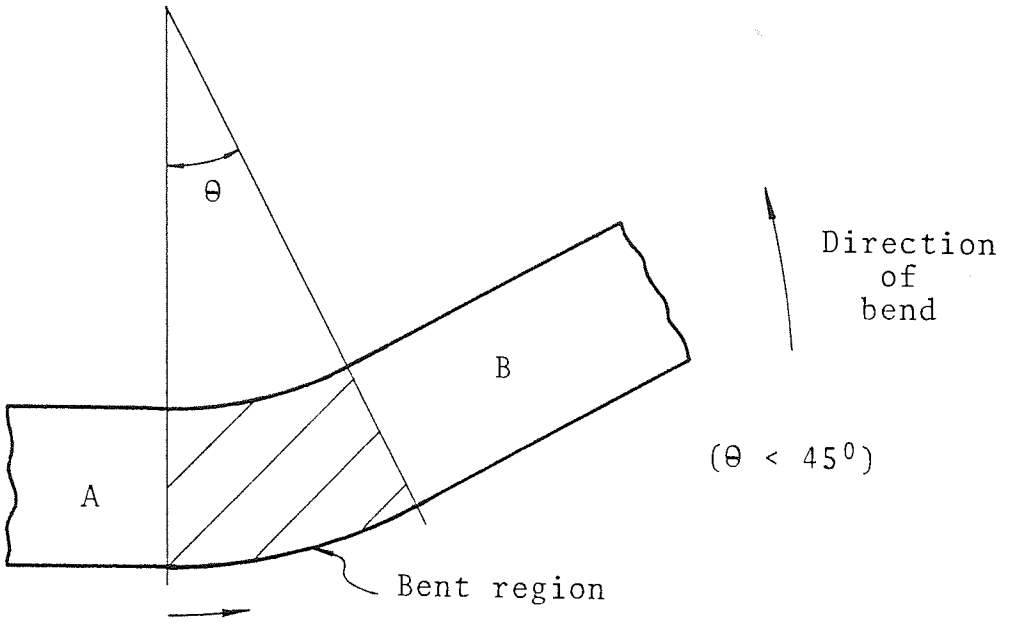


Fig 8.1 MATERIAL MOVEMENT DUE TO BENDING WITH ONLY TOP AND BOTTOM ROLLS

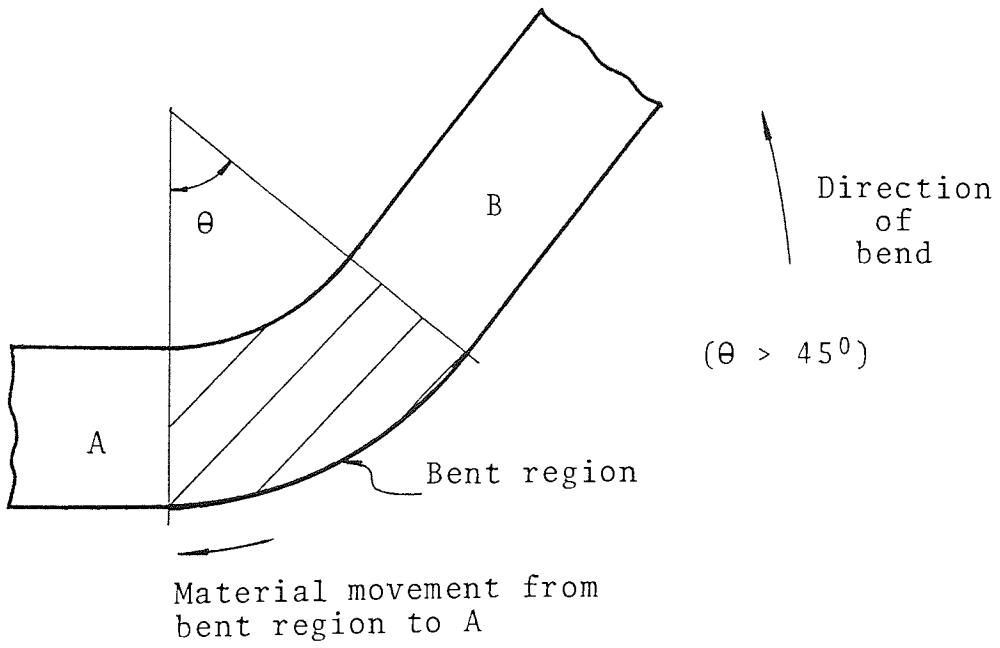
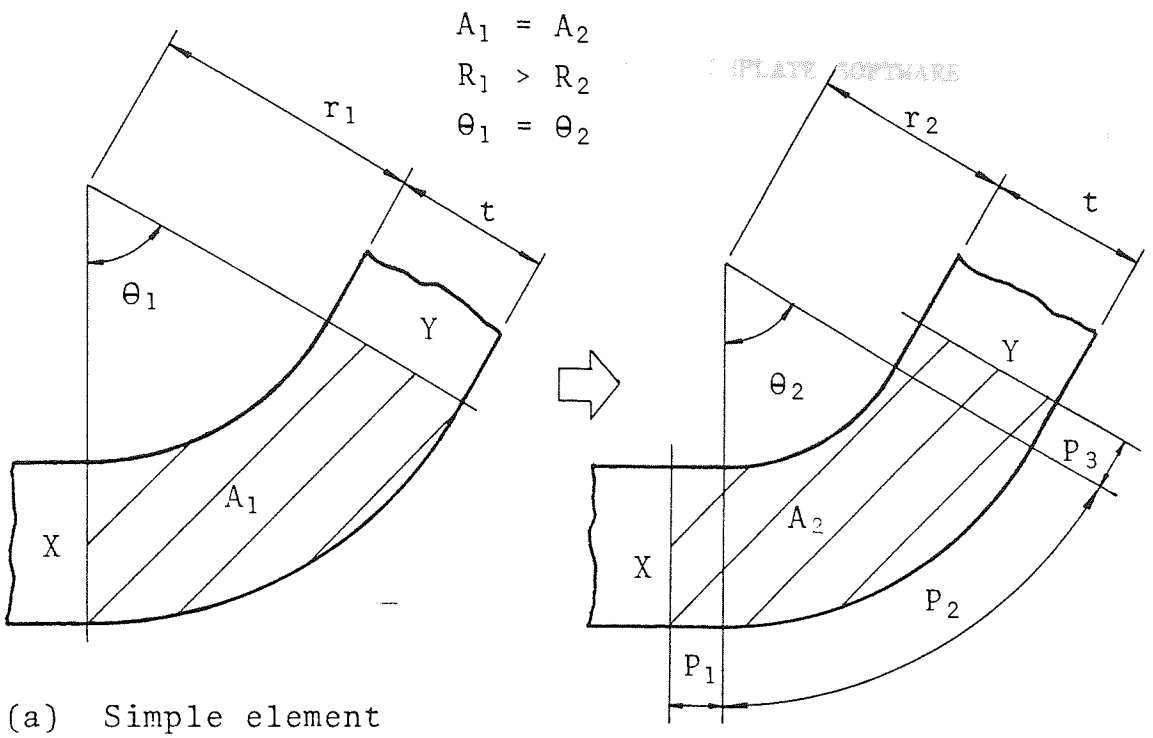


Fig 8.2 MATERIAL MOVEMENT DUE TO BENDING WITH SIDE-ROLLS



(a) Simple element

(b) Composite element

Fig 8.3 COMPOSITE ELEMENT LENGTH DEFINITION USING PERCENTAGES

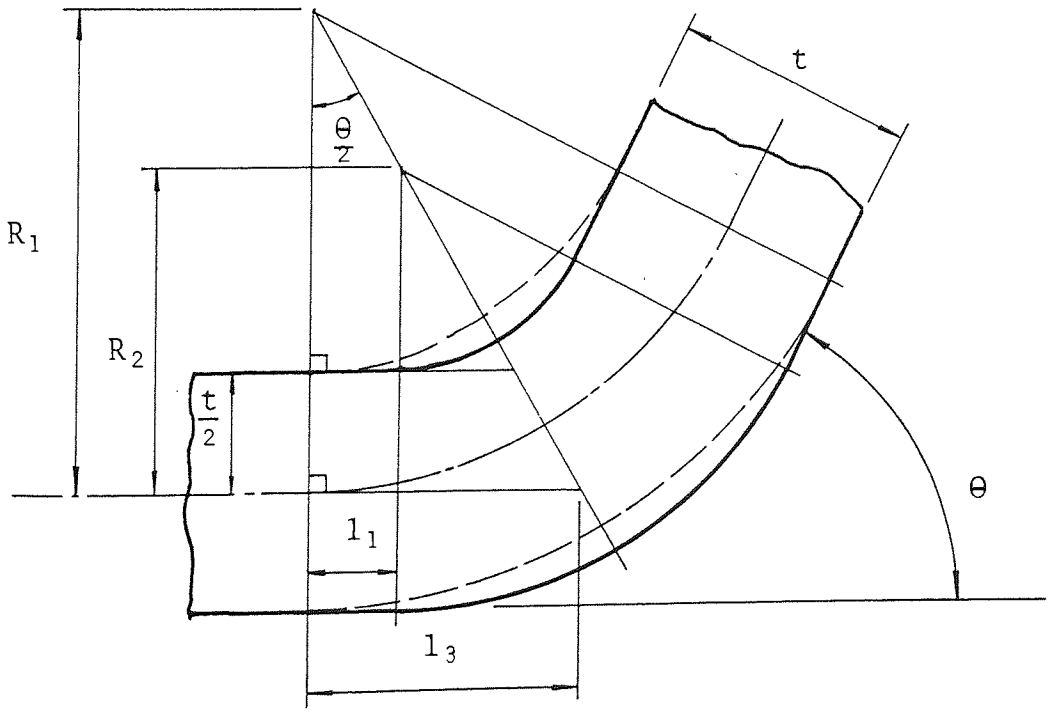


Fig 8.4 COMPUTATION FOR RADII-SHARPENING

TABLE 8.1 SUBROUTINES OF THE 10 TO 1 TEMPLATE SOFTWARE

NAME	LEVEL	FUNCTION
RPLOT3	1	Calls the subordinate subroutines to decode input data for defining element bending and element length monitoring, to compute and draw the 10 to 1 template shape at each bending stage and to output the corresponding template contour data for further processing.
RDPL	2	(see Table 7.1)
DCOFR	2	(see the Flower Pattern Input Decoder).
CTEM3	2	Initializes the data stores and controls the computation and drawing of the 10 to 1 templates for all stages.
STGTM	3	Updates the bending status or angles of bend of all elements for the current bending stage.
CLENG	3	(see the Element Length Monitor).
PLOTT	3	Controls the generation, drawing and output of templates.
STL1	4	(see Table 6.3)
ARC1	4	(see Table 6.3)
FRLM1	4	(see Table 6.3)
PLACT	4	Computes the maximum permissible scale for the template drawings as limited by the paper width and determines whether to use the ten times scale.
PLOT1	4	(see Table 6.3)

TABLE 8.1 SUBROUTINES OF THE 10 TO 1 TEMPLATE SOFWARE CONT/D ...

NAME	LEVEL	FUNCTION
TITLE	4	Prints the various headings for each template drawing.
OPTEM	4	Outputs the template contour data for each bending stage.
MAXMB	5	(see Table 6.3)
OPTBF	5	Generates either top or bottom template contour data for output.
MAXMI	6	(see Table 6.3)
OPTF	6	Generates the half-section template face for output.
OPRD	7	Outputs the data associated with each element contour.

TABLE 8.2 SUBROUTINES OF THE ELEMENT LENGTH MONITOR

NAME	LEVEL	FUNCTION
CLENG	1	Calls relevant subroutines for monitoring length definitions of left-hand side and right-hand side elements at the current bending stage.
STGLN	2	Distributes the element lengths according to the percentages defined via input for the composite circular elements. Linear parts of each composite element are added to the adjacent elements to keep a balanced sum of all the element meanlengths.
STGCP	2	As subroutine STGLN, but with fixed percentages, being 50% for linear parts and 50% for circular part.
STGSH	2	Adjusts element lengths and radii for radii sharpening.

CHART 8.1 HIERARCHY OF THE 10 TO 1 TEMPLATE PROGRAM (RPLOT3) (PART 1)

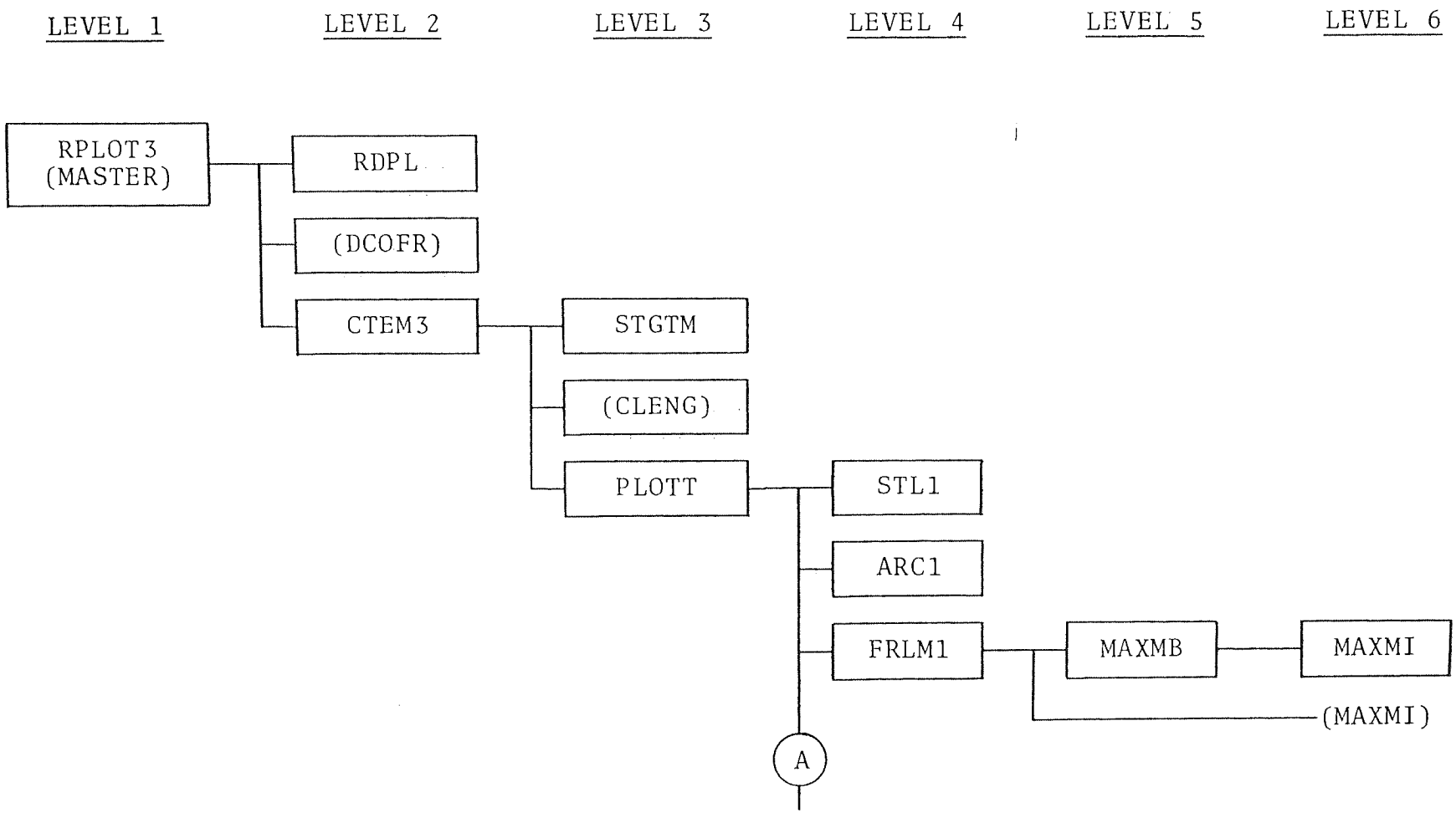


CHART 8.1 (PART 2)

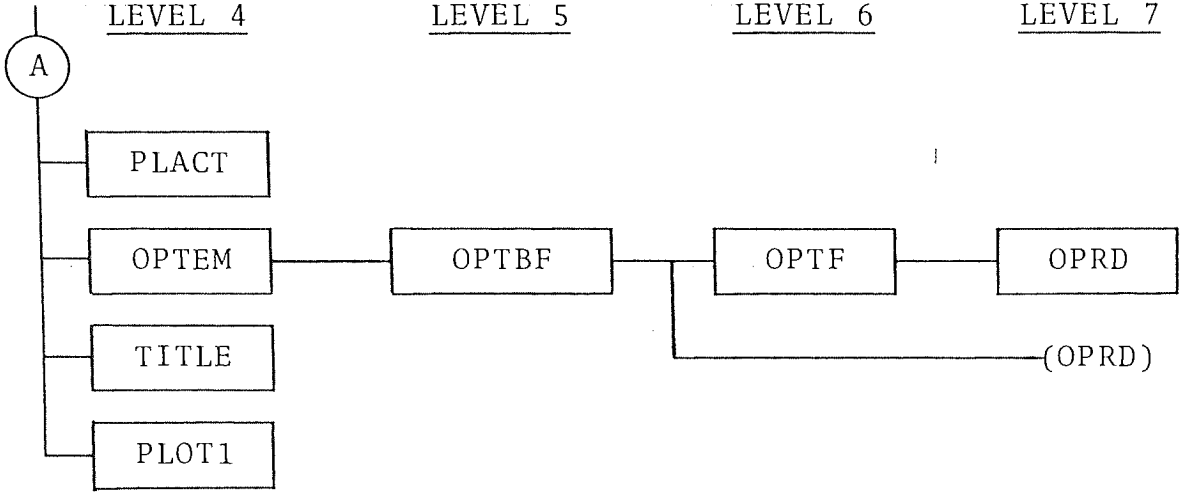


CHART 8.2 HIERARCHY OF THE ELEMENT LENGTH MONITOR (CLENG)

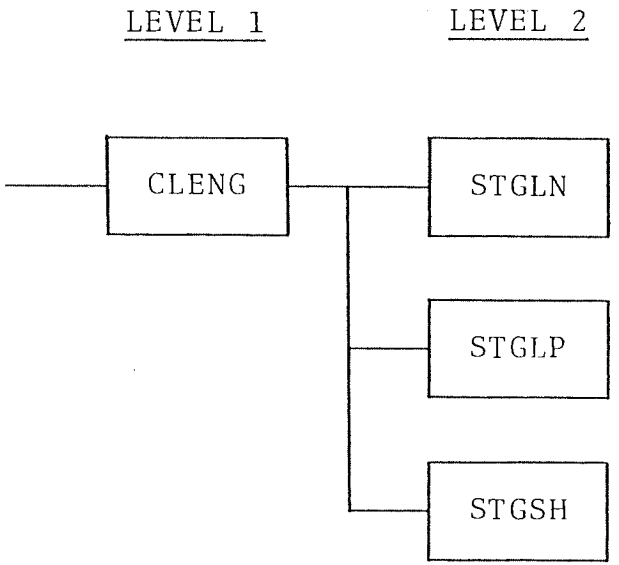


CHART 8.3 LOGIC FLOW OF THE 10 TO 1 TEMPLATE PROGRAM (RPLOT3) (PART 1)

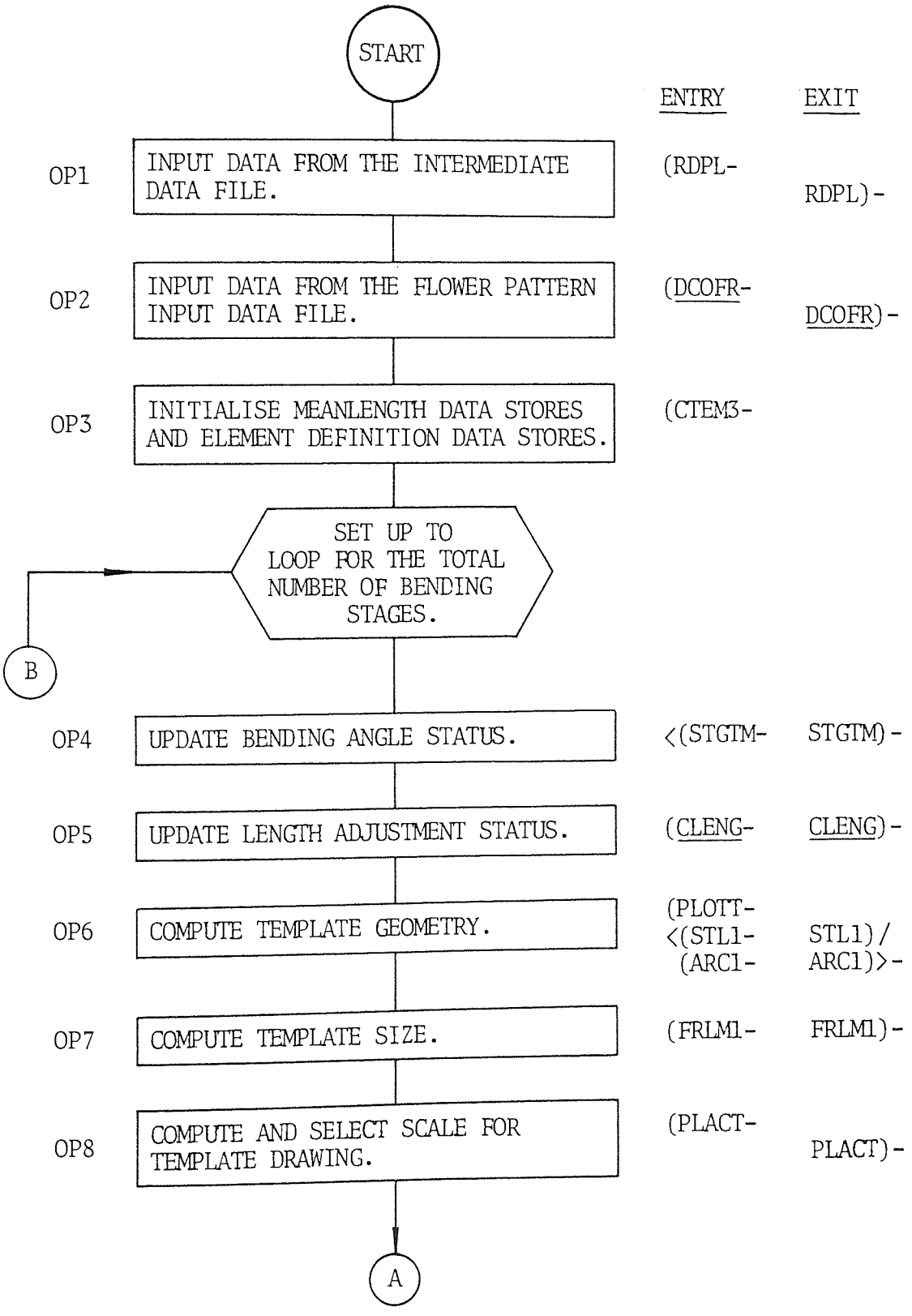
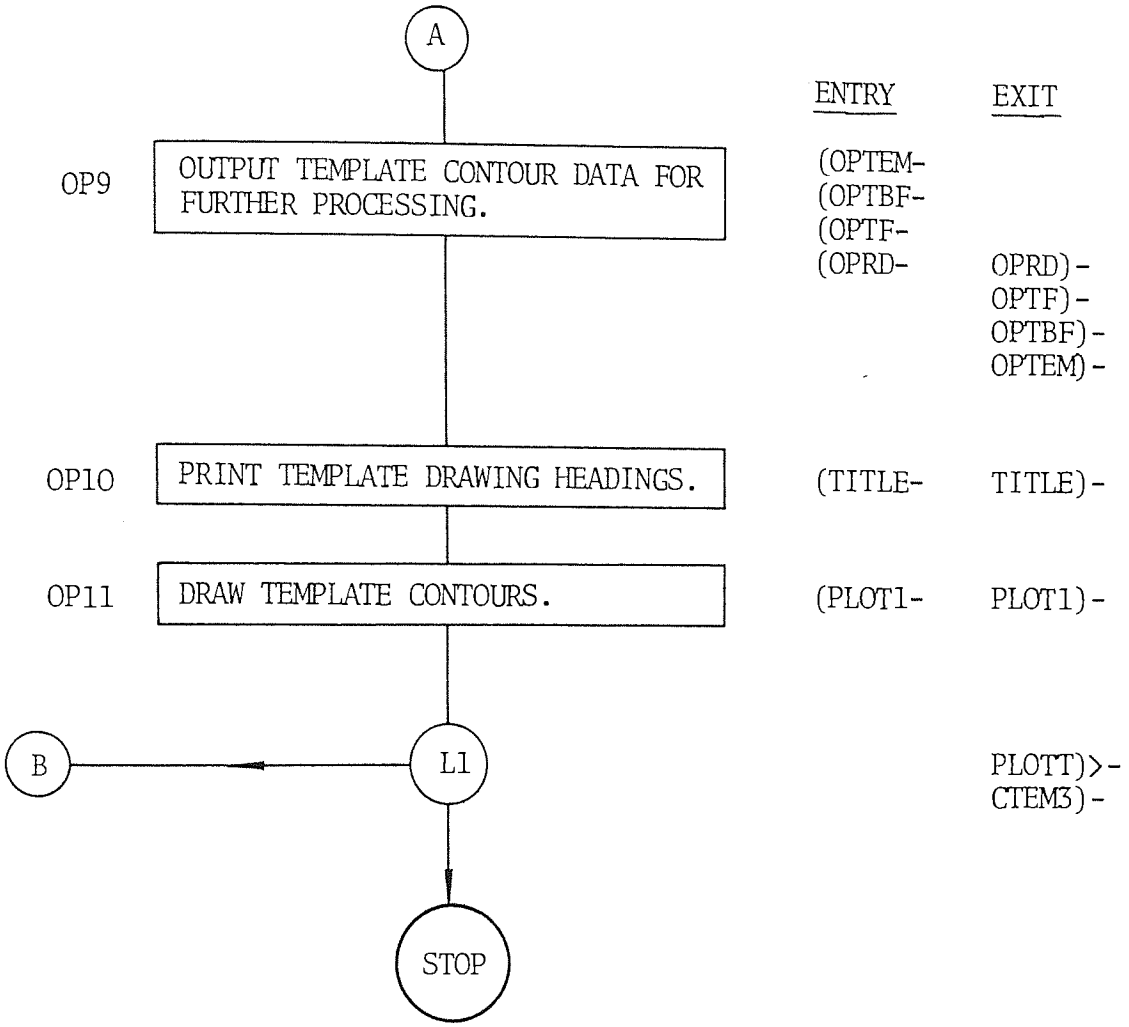


CHART 8.3 (PART 2)



CHAPTER 9

THE ROLL DESIGN SOFTWARE

9.1 Introduction

Existing practice in roll design and manufacture in the company has been to rely mainly on the use of wire-templates. Detailed drawings describing the precise roll contour geometry have not been generated or used, only simplified drawings indicating overall roll dimensions and clearances at various parts of the roll contours have been made available as a guide. All the time the machinists are expected to use the wire-templates as the master contour and to try to generate the roll contours such that the wire-templates would fit. A lot of skill in feeling the right kind of fit is involved rather than exact definition and measurement of each roll contour with precise figures. If automatic machines such as NC lathes are to be employed in future, such skill will not help, because now exact roll contour information will be required. Before that can be accomplished, acceptable roll-designs in a complete form suitable for manufacture must be generated first. A discussion of the work involved in computerising the process of generating the required roll design data which can be used subsequently in automatic NC manufacture is indicated by the following areas:

(1) BASIC ROLL CONTOUR DESIGN

(2) PINCH-DIFFERENCE SURFACES OPTION

(3) SIDE-ROLL OPTION

(4) EXTENSION-CONTOUR OPTION.

The fundamental area is of course the production of basic roll designs which are based on the template geometry. However, the other areas are also important because they too are essential features of complete working roll designs.

9.2 Basic Roll Contour Design

9.2.1 Introduction

In order to bend a section to the desired transitional shape identical to the shape of the wire-template at a particular bending stage, a pair of rolls should have contours forming a gap resembling the section shape itself and only through this gap the material will pass. In certain cases, the roll contours actually follow the template shape exactly (Fig. 9.1) but unfortunately, more often this is not the case (Fig. 9.2). As shown in Fig. 9.2, part of the roll contour following the template shape has to be modified to prevent roll material interfering with the strip material when the roll turns. This in fact constitutes one of the major problems in computerising the roll designing process. Despite the problems, roll contours in general do follow the template shape to a certain extent and that justifies the use of

template contour data as a basis for generating roll contour data.

One may appreciate that unlike the human brain and vision, which can perform complex pattern recognition and carry out the necessary modification process relatively easily with mere visual inspection and some thinking, the computer is not capable of such vision and such thought. Consequently such intelligence must be built into the computer via software programming to instruct the computer to recognise as much possible shape variation as feasible and to execute appropriate modifications. Depending upon the complexity of the shapes and the kinds of shape variation, the size and complexity of the software required varies. In general, it is not always viable or even possible to produce perfect software to cope with all situations, hence a compromise must be reached and normally could be reached in balancing sofware complexity against the range of shapes. Such has been the approach adopted throughout the development of the roll design software.

9.2.2 The Task

In simple terms, the process of generating roll contours based on the template contour information may be illustrated in the form shown in Fig. 9.3.

The whole process can be broken down into the following basic steps:

(1) To convert the template contour data to a form suitable for contour modification purposes.

(2) Repeatedly scanning the data store to identify areas requiring contour modifications.

(3) To perform the required contour modifications.

(4) To convert the modified contour data to a form suitable for output and further processing.

The main aspects carefully considered for designing and developing the software to peform the required task are summarised as follows:

(1) To create a working model which adequately defines the roll contour and which facilitates modifications.

(2) To design a data structure which holds all the necessary information representative of the roll contours before, during and after modifications.

(3) To establish the criteria which are associated with the shape orientation of one contour element with respect to the other and from which certain identifiable conditions may be deduced.

(4) To establish the required sets of conditions associated with each kind of shape peculiarities that require contour modifications.

(5) To establish the corresponding kinds of modification as required under each different set of

conditions identified.

(6) To devise means of data packing and repacking which are necessary for different modification requirements.

A new data structure was required for the roll contour data because the existing template contour data structure was not suitable for contour modification purposes. Repeated scanning of the data stores was required because normally more than one kind of modification is necessary for the same contour using the same data stores. In that way, only very occasionally the data has to be repacked or restructured to facilitate special modification procedures. However, to avoid conflict between each modification procedure during repeated scanning, the progress of each procedure has to be carefully monitored. At some stage data conversion and repacking were inevitable either to simplify the modification procedures or to suit different usage. Also as discussed in section 4.6, the set of conditions, decision rules and actions (in this case, modifications) have to be definable before computerisation is possible.

9.2.3 The Roll Contour Model

Like the template contours, a roll contour also consists of linear and circular elements. During contour modification, part or whole of the linear or circular element contours may have to be removed and replaced by a new contour (Fig. 9.4). The new contour is always linear and should be

vertical to avoid roll interference with the material being formed. The working model of the roll contour should therefore contain data which represent the template contour at the start, which can then be removed or modified and which can permit new data defining the new contour to be added.

Before the roll contour can be altered, it is necessary to locate exactly which parts are to be modified. To do that, conditions peculiar to the parts must be established so that they are identifiable, based of course on how each element is being placed in relation to the others in the contour, as a result of different bends, lengths and radii that form the peculiar types of shape. The first requirement is therefore some system which can keep track of the relative orientation and position of each element. An obvious candidate for this purpose is the cumulative angle technique (see section 6.3.1). Using the same technique each element orientation can be defined by a cumulative angle value while each element position and size can be defined by its absolute Cartesian coordinates in X and Y (Fig. 9.5). Here the only difference between linear and circular elements is that the linear elements have constant orientation whereas the circular elements have consistently varying orientation. That effectively means it is difficult to pin-point exactly what orientation a point (P_i) on the circumference of a circular element may have. Without knowing the exact orientation of these intermediate points, as shown in Fig. 9.6, problems arise when a point (P_i) on the circular element contour other than the start-point (P_1) and end-point (P_2) is involved in

the modified region of the roll contour. In such cases, the orientation of the intermediate point (P_i) must be known so that the need to modify the roll contour could be identified and the required modification could be done. Some means of defining the orientation for such points are therefore necessary.

A simple way for continuously tracking the orientation of the circular elements is to use linear envelopes to contain the circular contours (Fig. 9.7). That effectively enables the whole roll contour to be represented entirely by linear element contours. There are two main advantages associated with such form of representation. Firstly the contour modification procedures are greatly simplified since there is only one type of contour to deal with and it is relatively easy because the contours are linear. Secondly there is no need to create a special data structure to store the new contours which are to be inserted as a result of contour modifications. What is needed is only a circular contour to linear contour conversion process before the execution of all modification procedures and a revised process after that. The conversion process used is described as follows:-

To effectively enclose a circular element, it was found that tangent envelopes for the convex part and chord envelopes for the concave part were most suitable. Depending on the magnitude of the angle of bend at the circular element, there are four possible cases to consider for the enveloping operations (Fig. 9.7).

The computation of the angle for various orientations and the computation of the change-points for the enveloping linear contours are as follows:-

Points O, P_1, P_n, Q_1 and Q_{n-1} have their coordinate values obtainable directly from the template contour data store. The technique of deriving the coordinates of new points from the established points using equations 6.23 and 6.24 are again applied here. Using the centre point O as the reference point, and by working out the various distances and angles from point O to the change-points for the enveloping linear contours, namely the P points and the Q points, the coordinates of these points can then be computed, with reference to Fig. 9.7, as follows:-

Let r = inside radius of the circular element

t = thickness of the element

θ = angle of bend of the element

θ_m = cumulative angle of inclination at P_1

θ_{m+1} = cumulative angle of inclination at P_n

n = total number of convex points (P points).

Then the distance (R) between point O and the Q points is:-

$$R = r \qquad (9.1)$$

The distance (R_1) between point O and points P_1 and P_n is:-

$$R_1 = r + t \tag{9.2}$$

The distance (R_{1a}) between point O and points P_2 to P_{n-1} is:-

$$R_{1a} = \frac{R_1}{\cos(\theta_{n-2}/2.0)} \tag{9.3}$$

By carefully grouping all similar computation operations in the four cases of different θ values, a common procedure which applies to any of them has been constructed. For instance, point P_2 in case 1 can be derived in exactly the same manner as point P_3 in case 2, or point P_4 in case 3, or point P_5 in case 4. Note that the number of concave points (Q points) is always one less than the number of convex points (P points).

For cases 1, 2, 3 and 4, the (n-1)th angles of inclination of the change-points with respect to point O are

(1) Convex point (P_{n-1})

$$ANG1_{n-1} = \theta_{m+1} - \frac{\theta_{n-2}}{2} + 90^o \tag{9.4}$$

(2) Concave point (Q_{n-2})

$$ANG2_{n-1} = \theta_{m+1} - \theta_{n-2} + 90^o \tag{9.5}$$

For cases 2, 3 and 4, the (n-2)th angles of inclination of the change-points with respect to point 0 are

(1) Convex point (P_{n-2})

$$ANG1_{n-2} = ANG2_{n-1} - 45^{o} \qquad (9.6)$$

(2) Concave point (Q_{n-3})

$$ANG2_{n-2} = ANG2_{n-1} - 90^{o} \qquad (9.7)$$

For cases 3 and 4, the (n-3)th angles of inclination of the change-points with respect to point 0 are

(1) Convex point (P_{n-3})

$$ANG1_{n-3} = ANG1_{n-2} - 90^{o} \qquad (9.8)$$

(2) Concave point (Q_{n-4})

$$ANG2_{n-3} = ANG2_{n-2} - 90^{o} \qquad (9.9)$$

For case 4, the (n-4)th angles of inclination of the change-points with respect to point 0 are

(1) Convex point (P_{n-4})

$$ANG1_{n-4} = ANG1_{n-3} - 90^{o} \qquad (9.10)$$

(2) Concave point (Q_{n-5})

$$ANG2_{n-4} = ANG2_{n-3} - 90^{o} \qquad (9.11)$$

The coordinates of the change-points are then calculated by applying equations 6.23 and 6.24:-

(1) Convex points (P_2 to P_{n-1})

$$X_{n-j} = X_o + R_{1a} \cos(ANG1_{n-j}) \qquad (9.12)$$

$$Y_{n-j} = Y_o + R_{1a} \sin(ANG1_{n-j}) \qquad (9.13)$$

(2) Concave points (Q_2 to Q_{n-2})

$$X_{n-j} = X_o + R_1 \cos(ANG2_{n-j}) \qquad (9.14)$$

$$Y_{n-j} = Y_o + R_1 \sin(ANG2_{n-j}) \qquad (9.15)$$

where $j = 1, 2, 3$ or 4 ($j \neq 0$ and $j \neq n-1$)

9.2.4 Insertion of Linear Contour with Vertical Orientation

If a circular element happens to be at an orientation as shown in Fig. 9.8(a), with the existing roll contour model arrangement that uses linear enveloping contours, one of the change-points (M) will be wrongly used for contour modification purposes, resulting in the creation of a wrong modified contour (MN). In order that the correct modified contour (PQ) as shown in Fig. 9.8(b) will be created instead, the vertical

linear contour (PR) must be inserted before the roll contour modification operations. Effectively, as shown in Fig. 9.9(a) and (b), point M is eliminated and instead of the old enveloping contours (OM and MS), the new enveloping contours (OP, PR and PS) will be created such that the inserted contour (PR) is with a vertical orientation (Q_2). It is possible that a circular element may require both left-hand side and right-hand side inserts since the angle of bend for circular elements has been allowed to be greater than 180 degrees, requiring a total number of eight change-points instead of six in the original roll contour model.

Two separate procedures are needed for detecting conditions which need the vertical enveloping inserts, one for the elements with positive upward bend and one for the elements with negative downward bend. By normalising the angle of orientation (θ_s) of the start-point to a value within the range -360^o to $+360^o$, it is possible to reduce the total number of vertical orientation conditions, based upon the relative values of θ_s and θ_e, to only four for each procedure, as shown in tables 9.1 and 9.2. The corresponding modifications to the enveloping contours can then be carried out.

9.2.5 Roll Contour Data Structure

In order to preserve the unique identity of each element of the section, all information representing and related to the roll contour model which uses linear enveloping contours is stored on an element by element basis. Such a

measure is necessary because after all contour modifications have been performed, the circular element contours have to be restored from their linear enveloping equivalents.

Like the data structure for the finished section and the template contours, the roll contours are stored in units of top or bottom faces and left or right half-sections. For one-sided sections (ORIGIN is 1 or 2), only two stores are needed whereas for two-sided sections (ORIGIN is 3, 4 or 5), four stores are used. For processing contour data held in these stores, only one common set of procedures dealing with one store at a time is required instead of four separate sets, thereby saving 75 percent of the software space needed for processing purposes.

Each unit of data store for each element is further divided into two substores. The first substore ROXSX, as shown in table 9.3, is an array holding all information about the element identity and status, plus all reference data related to the element. The second substore ROXAX, as shown in table 9.4, is an array holding all information about the roll contour change-points, including their positions, orientations, types and status of existence. The change-points are used to represent the constituent linear contours of the roll contour model.

The roll contour data structure has been designed to be open-ended to accommodate future extensions without widely affecting the functions of the existing software.

9.2.6 Basic Roll Contour Modification

Careful study of the patterns of the roll contour has revealed that there are two major cases which require contour modification. To illustrate them, consider only the right-hand half-section of the top face contour in the following discussions:-

(1) The Case of Initial Upward Bend

Consider Fig. 9.10(a) and (b), both contours though looked quite different do show the same kind of orientation progression, namely that the cumulatively angle value of each constituent contour as one starts tracing from the origin O along the template face contour undergoes a definite pattern of change, in the following order:-

$$\theta_1 < +90^o$$

$$\theta_2 = +90^o$$

$$\theta_3 > +90^o$$

$$\theta_4 = +90^o$$

$$\theta_5 < +90^o$$

That effectively constitutes the following condition:-

"If the cumulative angle value of the contour first exceeds $+90^o$ and then it decreases to $+90^o$ or less, then the modified contour to be inserted will be from point P, just when the angle value decreases to $+90^o$, to a point Q vertically

below P, namely a point of equal X coordinate".

All element contours enclosed by line PQ are inaccessible from the top and therefore must be removed. The resulting contours are as shown in Fig. 9.11(a) and (b).

(2) The Case of Initial Downward Bend

For contour modification of the second kind, as shown in Fig. 9.12(a) and (b), a similar pattern applies, but the condition is based on different cumulative angle values:-

$\theta_1 > -90^o$

$\theta_2 = -90^o$

$\theta_3 < -90^o$

$\theta_4 = -90^o$

$\theta_5 > -90^o$

The corresponding condition is:-

"If the cumulative angle value of the contour first becomes less than -90^o and then it increases to -90^o or more, then the modified contour to be inserted will be from a point P, just when the angle value first becomes -90^o, to a point Q vertically below P, namely a point of equal X-coordinate".

All element contours enclosed by line PQ are inaccessible from the top and therefore must be removed. The resulting contours are as shown in Fig. 9.13(a) and (b). Note that this time point Q is further away from the origin

O than point P.

The modification procedure for both cases requires eliminating all the inaccessible element contours enclosed by line PQ and creating line PQ to replace them. Line PQ is then stored as an additional contour in the array of the element the end-point of which is Q. As all element contours are linear, the computation of the coordinates of point Q is simple, as illustrated below:-

In Fig. 9.14, the coordinates of the start-point $A(X_A,Y_A)$ and end-point $B(X_B,Y_B)$ of the element contour AB on which point $Q(X_Q,Y_Q)$ lies are known, so is point $P(X_P,Y_P)$, then point Q may be derived as follows:-

Since line PQ is vertical,

$$X_Q = X_P \tag{9.16}$$

By taking proportional distances in X and Y component directions:-

$$\frac{X_Q - X_A}{X_B - X_A} = \frac{Y_Q - Y_A}{Y_B - Y_A}$$

$$\text{or } Y_Q = Y_A + (Y_B - Y_A)\frac{X_Q - X_A}{X_B - X_A}$$

$$(X_B - X_A \neq 0) \tag{9.17}$$

$$\text{or } Y_Q = Y_A \quad (X_B - X_A = 0) \tag{9.17a}$$

It is important to note that in the developed software, contour modifications for the initial upward bend cases is

performed before those for the initial downward bend cases, more will be discussed in the next section.

9.2.7 Complicated Cases of Roll Contour Modification

The modification procedures decribed in the previous section is only adequate for the single-turn situations, if there are multi-turns in the vicinity of the modified zone, then complications will arise and the modified point Q will have to be selected correctly (Figs. 9.15 and 9.16). Again both the initial upward bend cases and the initial downward bend cases have to be considered separately.

(1) The Initial Upward Bend Cases

Fig. 9.15 illustrates various possibilities for the complicated multi-turns, in each case the correct point for the modified contour should be point Q though there are other points of identical X-coordinate value present (A and B).

In order to distinguish point Q from its invalid counterparts, additional identification conditions are needed as follows:-

(a) Point Q must be a point on that part of the contour facing upwards, corresponding to descending X-coordinate value as one scans backward along the contour from point P towards the origin O, thereby eliminating all points on parts facing downwards.

(b) Point Q must have a Y-coordinate value less than

that of point P, thereby eliminating all the invalid points vertically above point P.

(c) Point Q must be the last of all points on parts facing upwards as one scans backward from point P towards the origin O, thereby leaving the point nearest to P in the vertical direction.

(2) The Initial Downward Bend Cases

A similar dilemma exists with the cases of initial downward bend as shown in Fig. 9.16. Fortunately, if contour scanning and modifications for the initial upward bend cases are carried out before dealing with the initial downward bend cases, the sources of trouble will be eliminated with a replacement contour RS, hence no further action is necessary.

9.2.8 Replacing Roll Contours into Top and Bottom Halves

The roll contour model and the contour modification procedures discussed so far handle only contour accessible by roll material from one side, either top or bottom (Fig. 9.17).

Supposed those contour elements forming part of one face of the template happen to face the opposite roll rather than the roll on the same side, complications will result (Fig. 9.18) as part of the template contour will need to be detached from the roll contour store and repacked into the opposite roll contour store, in an infallible way. A suitable

way is to locate the extreme-X point, namely the point which has the highest X-coordinate value with respect to the origin O, and use the point to split the template contour into the two required parts. The part belonging to the opposite side is then detached and repacked into the opposite roll contour store in reversed order and orientation.

The data values associated with the affected part undergo the following conversion procedures to suit their new orientation requirements:-

(1) Clockwise bends become anti-clockwise bends and vice versa.

(2) Cumulative angles for the element contours are modified to give orientations exactly opposite to their original directions.

(3) The sequence of the points of each element contour is reversed so that end-point becomes start-point and vice versa.

If the extreme-X point occurs in an element, usually a circular element, the element is split up into two shorter elements, one is retained in the original store to replace the original element with the extreme-X point as the new end-point and the other is transferred along with other elements following it in the sequence to the opposite roll contour store.

Repacking has to be and is carried out just before

entering the contour modification procedures.

9.2.9 Restoring Circular Element Contours after Modifications

Circular elements which do not have any of their constituent contours altered during contour modifications do not require any restoration, only those elements with parts removed and modified need restoration.

As an illustration, consider the case of the initial upward bend modification (see section 9.2.6) as shown in Fig. 9.19. The circular element AC has one of its enveloping contours BC modified to BD when the vertical insert QD is included during contour modification. However, the true end-point of the modified circular element should be point E instead of point D, so restoration of the true coordinates for the modified change-points of the circular element is necessary. As for the other end of the vertical insert the circular element PS has its start-point P removed and replaced by point Q, which also needs restoration since the true start-point is point T. Hence both the modified end-points and start-points of circular elements need restoration.

In the case of the initial downward bend (Fig. 9.20), the situation is similar except that the vertical insert is stored at the top element since the roll contour proceeds from top to bottom instead of the opposite way.

In both cases, in restoring the true start-point and

end-point of a particular element, the X-coordinates remain valid while only the Y-coordinates need to be adjusted, as follows:-

Let R = element contour radius

X_c = X-coordinate of the centre point

Y_c = Y-coordinate of the centre point

X = X-coordinate of the circumference point

Y = Y-coordinate of the circumference point

with the standard circle equation:-

$$(X - X_c)^2 + (Y - Y_c)^2 = R^2$$

And knowing R, X_c, Y_c and X, the value of Y can be derived:-

$$Y = Y_c \pm \sqrt{R^2 - (X - X_c)^2} \qquad (9.18)$$

The sign of the square-rooted part can be determined by the direction of bend, namely Y should be greater than Y_c if it is a clockwise (negative) bend and less than Y_c if it is an anticlockwise (positive) bend.

9.2.10 The Overall Process of Generating Basic Roll Contours

Summing up the procedures described so far for the generation of basic roll contours, the overall process may be

represented by the data flow diagram in chart 9.1.

9.3 The Pinch-Difference Option

9.3.1 Introduction

Pinch-difference is a quantity for controlling the variation of the gap size between opposing surfaces of a roll pair at different points across the strip width at a particular bending stage (Fig. 9.21). Such variation is necessary because with a constant rotational speed of the rolls, different roll faces which are at different radii from the roll axes do not have the same peripheral speed. If all the peripheral surfaces are in firm contact with the strip there will either be excessive friction between the roll and the strip or the strip may be dragged at peripheral surfaces of slower speed thereby causing distortion in forming, which can be disastrous. To overcome that, the normal practice is to have only a pair of surfaces which are parallel to the roll axes pressed firmly against the strip, while the other surfaces are forming the strip to the desired shape with only one side of the strip in contact with the roll and not the other side of it. The former are termed drive-surfaces while the latter are termed clearance-surfaces. With some shapes, there may not be a pair of horizontal surfaces available for driving the strip, in that case one of the points of horizontal orientation on the strip will constitute the drive-surfaces.

When setting up the rolls in a forming station, normally

the bottom roll is fixed in position while the top roll may be moved up and down for height adjustment. Such adjustment is necessary to ensure that the drive-surfaces of the rolls are firmly pressed against the strip because different batches of the strip material do not necessarily have the same thickness. The variation or tolerance can be as big as ± 8 percent of the material thickness.

9.3.2 Pinch-Difference Surfaces Design

A few methods have been used in practice to design pinch-difference surfaces to serve the required purpose:-

(1) METHOD 1

This is the manual method of working out each gap size at appropriate points using empirical formulas and tables. Typical tables used are as shown in Tables 9.5 and 9.6.

Based on the material thickness, the corresponding tolerance is obtained from Table 9.3 and from that the maximum thickness variation is derived:-

$$\text{MAXIMUM THICKNESS VARIATION} = |\text{POSITIVE TOLERANCE}| + |\text{NEGATIVE TOLERANCE}| \qquad (9.19)$$

The gap distance between drive-surfaces is taken to be the material thickness at minimum material condition:-

$$\text{MINIMUM LIMIT} = \text{NOMINAL THICKNESS} - |\text{NEGATIVE TOLERANCE}| \qquad (9.20)$$

The pinch-difference value is then obtained from Table 9.6 according to the Maximum Thickness Variation value and the angle of bend and with that the clearance-surfaces gap is derived.

CLEARANCE LIMIT = MINIMUM LIMIT +
(MAXIMUM THICKNESS VARIATION
- PINCH-DIFFERENCE VALUE)
+ ADDITIONAL ALLOWANCE (9.21)

This method is not suitable for computerisation since there are too many clearance-surfaces involved, which can lead to complications.

(2) METHOD 2

This method uses a constant clearance-surface gap and a drive-surface gap (Fig. 9.22).

The gap (D) at the drive-surface is always smaller than the gap (C) at the clearance-surface.

Usually the size of the two gaps are determined as follows:-

The clearance-surface gap,

C = Thickness at Maximum Material Condition +
Clearance (9.22)

The drive-surface gap,

D = C - constant (9.23)

Normally by taking a maximum tolerance of $\pm$10% of thickness for the material and a constant value 0.07mm (or 0.003in), it is adequate to obtain the required pinch-difference clearances for forming.

This "double-thickness" method is a more suitable method for computerisation since only two kinds of surfaces are involved and they are adequately defined.

(3) METHOD 3

This method uses only one constant clearance surface gap (Fig. 9.23).

The gap (C) is determined by

C = Thickness at Maximum Material Condition
+ Clearance (9.24)

As the gap is larger than the maximum material thickness, the top-roll has to be lowered vertically until its horizontal surface is in contact with the strip surface, the other non-horizontal surfaces will still be in the clear. This method is good for rolls with only one pair of horizontal surfaces. It is not suitable for rolls with more than one pair of horizontal surfaces or with only slant surfaces.

This "single-thickness" method can also be easily computerised since only one kind of surface is involved.

Methods 2 and 3 have therefore been chosen for

computerisation purposes. The difficulty involved was to incorporate them into the existing basic roll contour software.

9.3.3 The Pinch-Difference Surfaces Model

For computerisation purposes only two kinds of pinch-difference surfaces need to be considered, namely the drive-surface and the clearance-surface. Methods for defining these two different surfaces must be created so that designers can decide where the surfaces should be and what clearance value to use at different bending stages. Internally in the software facilities for identifying the different element contours belonging to each type of the surfaces and for making the corresponding thickness adjustment accordingly must also be established.

For simplicity in usage, a surface definition scheme requiring the designers to define only the starting element and the ending element of the drive-surface has been adopted. With that the drive-surface elements can be distinguished from the clearance-surface elements without ambiguity.

To avoid any undesirable interference with the existing software structure for generating roll contours, a system of flags have been used for identifying the drive-surface elements (Fig. 9.24). The flags in this case happened to be the seventh element of the ROXSX substore (see Table 9.3).

The pinch-difference surface model is however still

incomplete as can be seen in Fig. 9.24, there are three possible methods of distributing the pinch-difference clearance value at the transitional points (Fig. 9.25), method (a) uses an unbroken bottom surface, method (b) uses two broken surfaces while method (c) uses an unbroken top surface. As the template contour has been derived using the bottom surface as the datum surface it is desirable to keep it unbroken so as to avoid complications in datum readjustment from element to element once the pinch-difference surfaces are introduced. In practice, it is also better to keep the bottom surface smooth since the bottom roll is fixed and the strip sitting on an unbroken surface is less likely to wobble. As a result method (a) has been selected. If in future method (b) or method (c) is preferred, computation in the software may be modified accordingly (Fig. 9.26).

The next thing to consider is how to bridge the gap caused by the change of thickness at the transitional points. To maintain the surface continuity, a bridging contour must be created, Fig. 9.27 shows a few practical alternatives. Of the alternatives illustrated, the only one which does not cause complications in shape computation at the transitional point is the normal linear contour. Besides, as the Roll Contour Model itself is made up of linear contours, linear inserts are also easier to incorporate.

Having considered the various possibilities and made the required selections, the final pinch-difference surfaces model

(Fig. 9.28) constructed and used has the following summarised properties:-

(1) It generates only two kinds of roll surfaces, namely the drive-surface and the clearance-surface.

(2) It uses elements for distinguishing the drive-surface part from the clearance-surface part.

(3) The bottom surface is always continuous while the top surface may be broken at element boundaries to allow the required change in gap size.

(4) The inserted contour at each transitional point for joining the drive-surface to the clearance-surface is linear and is normal to the element orientation vector at that point.

9.3.4 Pinch-Difference Software Incorporation

It was necessary to select a suitable point in the roll contour data processing at which the pinch-difference surface processing might be incorporated without upsetting the existing software logic and leading to errors which might have been difficult to trace and resolve. The point selected was the part of processing which generates the template contour data; the original procedures were replaced by similar procedures which also perform the element contour computation but take into account the thickness adjustments for generating the required pinch-difference surfaces. To avoid interfering

with the existing template contour to roll contour conversion procedures, the software was designed to store the inserting contours at the transitional points separately and to incorporate them into the roll contour data stores after the conversion. Such measures had led to least interference to the existing software logic and structure and had made the pinch-difference surfaces software easily detachable and replacable and therefore truely optional.

9.3.5 Data Structure

As the circular element data store of the roll contour model may be fully occupied, the pinch-difference inserts at the transitional points will have to be added to only the linear element stores which use up only two of the eight available change-point area per element. The problem of possible absence of a linear element in between successive circular elements in the element definition sequence at the transitional points does not arise because the introduction of the composite element length definition option (see section 8.3) does not permit such absence.

Since the transitional point can occur either at the beginning or at the end of a linear element, two separate insert stores are required (Table 9.7).

9.3.6 Generating Pinch-Difference Surfaces

The pinch-difference surfaces, like the template contours,

are also generated in half-sections and in top or bottom faces. They are also built up element by element in ascending order of the element sequence numbers. The system flags for monitoring the surface generation are as shown in Tables 9.8, 9.9 and 9.10.

From the values of the surface mode flags (Table 9.8), the occurrence of a transitional point can be detected whenever there is a change of the element mode from drive-surface to clearance-surface as indicated by the identifier flags (Table 9.3). The insert flags are then set or cleared accordingly to indicate whether an insert contour is required. The gap size for the element currently being processed is also set up accordingly via the flag IDIR.

The computation procedures used for the elements of the finished section (see section 6.3) have been adapted to account for the change in thickness and the creation and storage of the insert contour at the transitional points. Virtually the same set of equations have been used for computing each element geometry but with suitable adjustment to the points and distances incorporated (Fig. 9.26), including the shift of the reference point as a result of change in the gap size if so required.

The insert contours at the transitional points, being stored separately, are incorporated into the roll contour data store when the process of template contour to roll contour conversion has been completed. No further pinch-difference

computation is required from that point onwards.

9.3.7 The Overall Process of Generating Roll Design Contours with Pinch-Difference Surfaces

The overall process of roll design contour generation incorporating the pinch-difference surfaces option is illustrated by the data flow diagram in chart 9.2.

9.4 The Side-Roll Option

9.4.1 Introduction

With only the top and bottom rolls, it is sometimes inadequate to form a required section shape satisfactorily. For instance, elements which are bent to almost vertical orientation are difficult to form with the top and bottom rolls alone, the use of side rolls is therefore necessary (Fig. 9.29). As side-rolls are indispensible in generating complete roll designs, a software option for generating side-roll contours has thus been developed.

9.4.2 Side-Roll Definition Scheme

Fig. 9.29 shows that, like the top and bottom rolls, side-roll contours are also parts of the template contour. In fact, they are those parts which do not form the top and bottom roll contours.

The complicated variation of the template geometry does

not permit precise description of a consistent set of identifiable circumstances which warrants the use of side-rolls in the roll-designs, hence automatic selection of side-rolls is impracticable. Reliable decisions on whether to select side-rolls in a given situation must still come from the designers themselves. The best aid to offer the designers is therefore one which performs the automatic generation of side-roll contours as specified and as defined by the designers. The following instructions are necessary for such purposes.

The designers are first expected to indicate which side-roll, left or right (or perhaps both) is required at a particular bending stage. Then they proceed to define each side-roll contour according to the following scheme.

The scheme consists of designating each possible face of an element with a particular face number with respect to the direction of the element definition sequence (Fig. 9.30) and specifying the beginning and the end of the side-roll contour in terms of the element faces. Faces 1 and 2 are always present but faces 3 or 4 is present only at the end elements (Fig. 9.31). As an example (Fig. 9.32), to define a side-roll contour AB, the designers are required to specify the beginning element face, in this case element 6 face 1, and the ending element face, in this case element 2 face 2. The side-roll contour defined is always assumed to start from the bottom upwards. The definition scheme works satisfactorily provided the following rules are observed:-

(1) RULE 1

The side-roll contour must be a continuous contour which consists of element faces accessible from the side. Obviously it is pointless to include in the side-roll contour element faces which cannot be reached from the side but unfortunately such mistakes cannot be detected during input as side-roll contour geometry is not available until much later in the program run.

(2) RULE 2

The side-roll contour must include the element face which contains the furthest point from the origin in the horizontal direction, namely the extreme-X point that separates the top roll contour from the bottom roll contour during repacking (see section 9.2.8). This is because if such element face is excluded when defining the side-roll contour, then part of the element face becomes inaccessible from either the top roll side or the bottom roll side (Fig. 9.33).

In certain cases, a bent element may be only partially accessible from the side and also contains the extreme-X point (Fig. 9.34(a)). By Rule 1 it should be excluded but by Rule 2 it should be included, there is obviously a conflict and dilemma. The only available solution is to split the element into two separate elements (Fig. 9.34(b)) such that one element face is completely accessible from the side and also contains the extreme-X point. The split-point can be

anywhere in the region accessible both from the side and from the top or bottom (Fig. 9.34(a)) and is therefore quite arbitrary. The job of splitting the element is thus left to the designers since the rules of fixing the split-point are undefined.

9.4.3 Side-Roll Contour Model

As side-rolls are also form-rolls, they have identical characteristics as their top and bottom counterparts. By making the side-roll contour model identical to the basic roll contour model, using the same data structures (Tables 9.3 and 9.4) but of a smaller size, the same set of procedures used for top and bottom roll contour processing can be used again for side-rolls, thereby avoiding the duplication of procedures and saving a considerable amount of software space.

The side-roll elements in the basic roll contour data stores are identifiable through a series of preset flags according to the side-roll contour definition supplied by users. These elements are detached from suitable parts of the stores in a definite sequence to form the required side-roll contours, either left or right, which are then converted to an orientation that appears like a bottom roll contour. The following is the required conversion procedure:-

Fig. 9.35 shows the relationship between the reference-axes for the top and bottom rolls and the pseudo reference-axes for the side-rolls appearing like bottom rolls. Every

point (Q) with respect to the new set of axes (X^1 - Y^1) can be mapped to the corresponding point (P) on the original set of axes (X - Y), so are the new angles (Ø) which are based on the old angles (θ). The following are the equations required:-

(a) For two-sided sections (i.e. ORIGIN is 3, 4 or 5) and the extreme-X end of the one-sided sections (i.e. ORIGIN is 1 or 2),

$$X_Q = Y_p - Y_s \tag{9.25}$$

$$Y_Q = L - X_p \tag{9.26}$$

$$\emptyset = \theta - 90^o \tag{9.27}$$

(b) For the starting end of the one-sided sections (i.e. ORIGIN is 1 or 2),

$$Y_Q = Y_p + Y_s \tag{9.28}$$

$$Y_Q = L - X_p \tag{9.29}$$

$$\emptyset = (180^o - \theta) - 90^o \tag{9.30}$$

Case (b) appears to be different because it incorporates lateral mirror-image transformation which has been used for all two-sided section computations and drawing.

The above conversion procedure uses data in the bottom roll orientation. If part of the top roll contour also

constitutes the side-roll contour, it first has to be converted to the bottom roll orientation before applying the procedure. The only complication occurs in the starting end of the one-sided sections, in which the top roll orientation is already in the correct order and it is the bottom roll contour which is to be corrected.

The following conversions are necessary to reshuffle either the top roll part or the bottom roll part in a bottom upwards sequence for constructing side-roll contours:-

(1) Clockwise bends becomes anti-clockwise bends and vice versa.

(2) Cumulative angles are changed to orientations directly opposite to their original directions.

(3) The sequence of the contour points for each element involved is entirely reversed so that the old end-points become the new start-points and vice versa.

Elements containing the extreme-X points are not split up if they are part of side-roll contours.

9.4.4 Side-Roll Software Incorporation

Like the pinch-difference option, the side-roll option has been incorporated with least interference to the logic and structure of the existing bulk of software so that it is

truely optional. The most appropriate point selected for conducting the side-roll contour processing is when the basic roll contours have been repacked into top and bottom halves before going into the contour modification process. At that point the side-roll parts of the basic roll contour are detached and set up in the respective side-roll data stores in the correct sequence for independent processing, only the top and bottom roll parts remain intact.

9.4.5 The Overall Process of Generating Roll Design Contours with Side-Rolls

The overall process of roll design contour generation incorporating side-rolls is as illustrated by the data flow diagram in chart 9.3.

9.5 The Extension-Contour Option

9.5.1 Introduction

With the roll contours based on template contours alone, the rolls produced have the tendency to slide along their axes of rotation creating forming problems. To prevent that from happening, extension-contours which interlock the matching rolls in position are used in practice. Such contours are not part of the template surface contours, they are instead contours extended from the break-points which separate one roll contour from the other. For the top and bottom rolls without side-rolls (Fig. 9.36(a)), a pair of extension-contours

on both sides of the rolls, called spigots, are used. If side-rolls are used (Fig. 9.36(b)), the extension-contours which still serve the same purpose are slightly different in shape, also there are two pairs of them instead of one. Facilities for defining and generating the extension-contours to be attached to the roll design contours have also been incorporated in the developed software.

9.5.2 Extension-Contour Definition Scheme

The following common types of extension-contours have been included in the definition scheme:-

(1) Spigots

There are two main types of spigots in use, as shown in Figs. 9.37 and 9.38. The first type consists of four parts whereas the second type consists of only three.

(2) Side-Roll Extension-Contours

Two pairs are required at the top roll side and the bottom roll side respectively, there are more variations in these contours as shown in Fig. 9.39(a) and (b). The number of parts in each case is different, the maximum is four.

In order to define both the spigots and the side-roll extension-contours, the following scheme has been created:-

All extension-contours are to be defined as any of the four types shown in Table 9.11, each consisting of four linear

parts with fixed orientation for each part. Parts which are not required may have zero value specified for their length if less than four parts are intended. Both left-hand side and right-hand side extension-contours have to be defined, only one pair if spigots are used and two pairs if side-rolls are present, at each bending stage.

9.5.3 Extension-Contour Scheme Incorporation

Like the other options described earlier, the extension-contour option has also been incorporated with least interference to the logic and structure of the existing software already developed, to make it completely optional. The appropriate point in the software execution for processing the extension-contours was found to be immediately after the side-roll contour processing. Two stages were required, the first establishes the break-points that separate one roll contour from the other and the second generates the extension-contours.

9.5.4 Generating Extension-Contours

Fig. 9.40 shows the break-points to be established depending on whether side-rolls are present. Like other roll contour processing, the extension-contours are generated in half-sections, either left or right. If side-roll is absent in the half-section, only one break-point which is the extreme-X point and one common extension-contour shared by the top and

bottom rolls are required, otherwise two break-points and two extension-contours must be used.

With the specified cumulative angle of inclination and the length associated with each extension-contour part (Table 9.11), the required change-points for each part of the extension-contour selected may be computed, again with equations 6.23 and 6.24. The extension-contour data are stored in the array EXCONX (Table 9.12). The break-points which are the starting points of the extension-contours are stored in the array BRKPTX (Table 9.13).

The angles for the inclined part of the extension-contour blending with the top roll and the bottom roll have been fixed at $+30^{o}$ and -30^{o} respectively. The angle of the inclined part for spigots normally follows the angle at the break-point, except when the break-point has a vertical orientation (Fig. 9.41(a)) it is then necessary to modify the angle to a horizontal orientation (Fig. 9.41(b)).

9.5.5 The Overall Process of Generating Roll Design Contours with Extension-Contours

The overall process of roll design contour generation incorporating extension-contours is as illustrated by the data flow diagram in chart 9.4.

9.6 The Program

The software or program for the generation of the roll

design drawings and data consists of four distinctive parts, namely:-

(1) The Roll Design Program (RPLOT4)

(2) The Flower Pattern Input Decoder (DCOFR)

(3) The Roll Input Decoder (DCORL)

(4) The Element Length Monitor (CLENG)

The Flower Pattern Input Decoder has been described in section 7.6.2 and the Element Length Monitor has been described in section 8.7.2. The Roll Input Decoder will be described in section 9.6.2.

9.6.1 Program Structure

The hierarchy of the program structure is shown in chart 9.5, with the main program RPLOT4 residing at the highest level of execution, controlling the sequence of execution of the subroutines at lower levels. A brief account of the nature of the subroutines is given in Table 9.14.

9.6.2 The Roll Input Decoder

This module is designed to deal with all input data from the Roll Input Data File (RXXXX). The hierarchy of the subroutines in this module is as shown in chart 9.6 and Table 9.15 describes the subroutines involved.

9.6.3 Program Logic Flow

The logic flow of the Roll Design Program (RPLOT4) is summarised in chart 9.7, using the Symbolic Major Logic Representation (SMLR) notations (see Appendix 2).

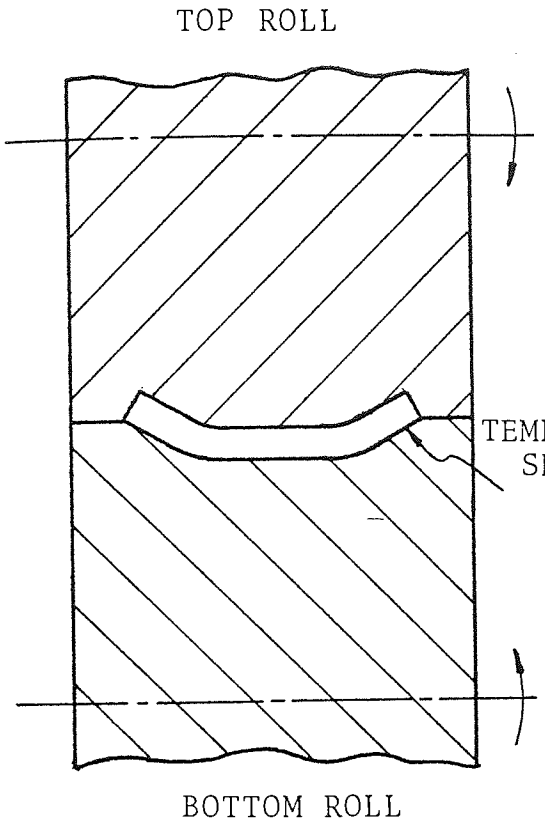


Fig 9.1 A CASE OF ROLL CONTOUR FOLLOWING THE TEMPLATE SHAPE EXACTLY

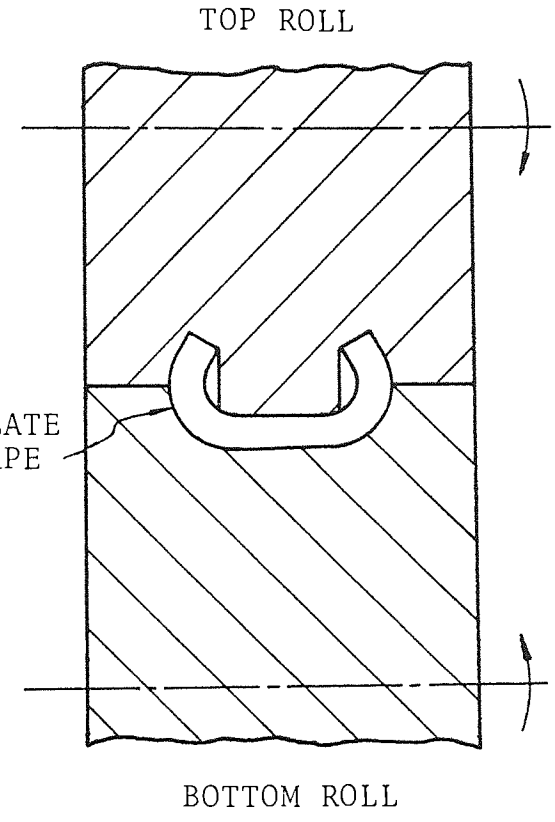


Fig 9.2 A CASE OF ROLL CONTOUR PARTIALLY FOLLOWING THE TEMPLATE SHAPE

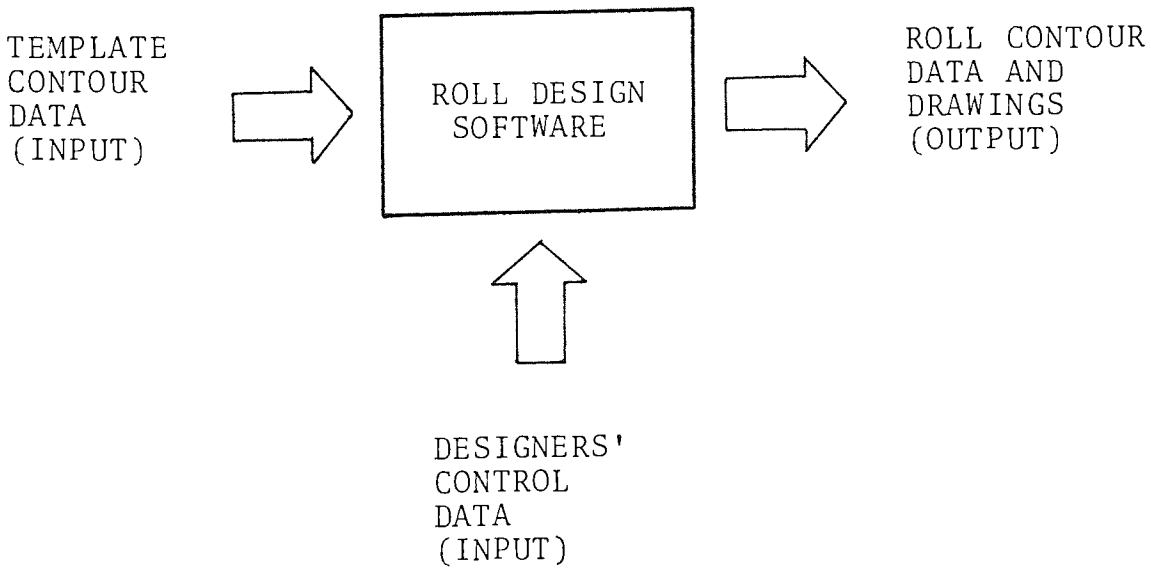


Fig 9.3 THE ROLL DESIGN SOFTWARE PROCESS

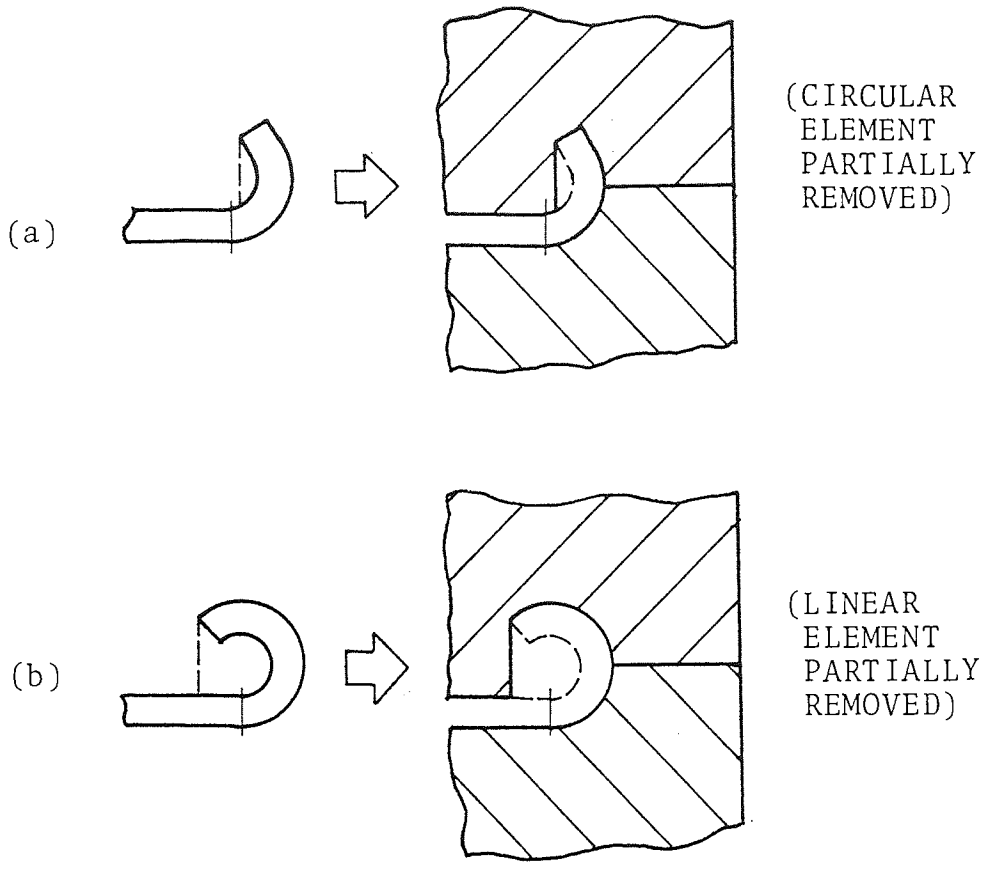
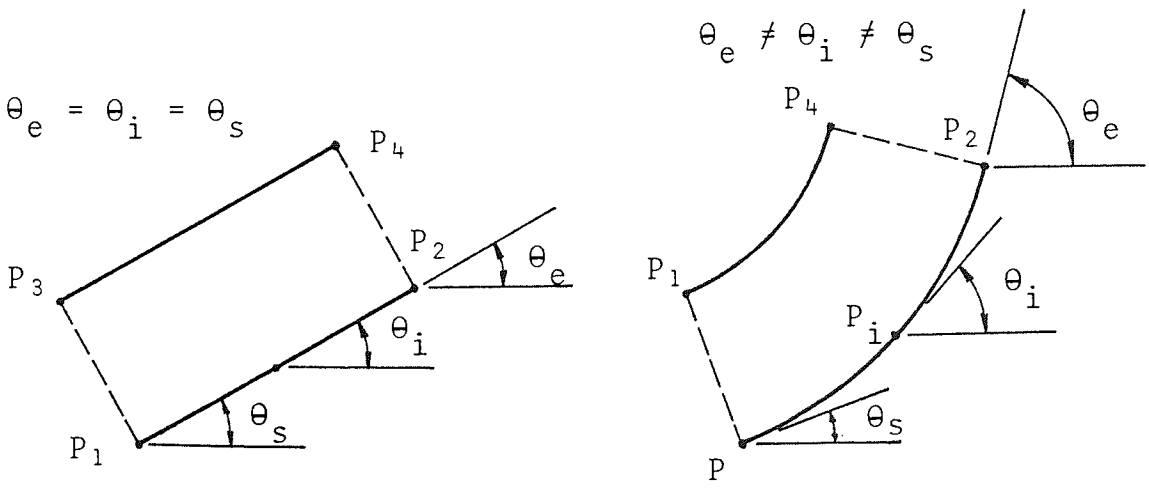


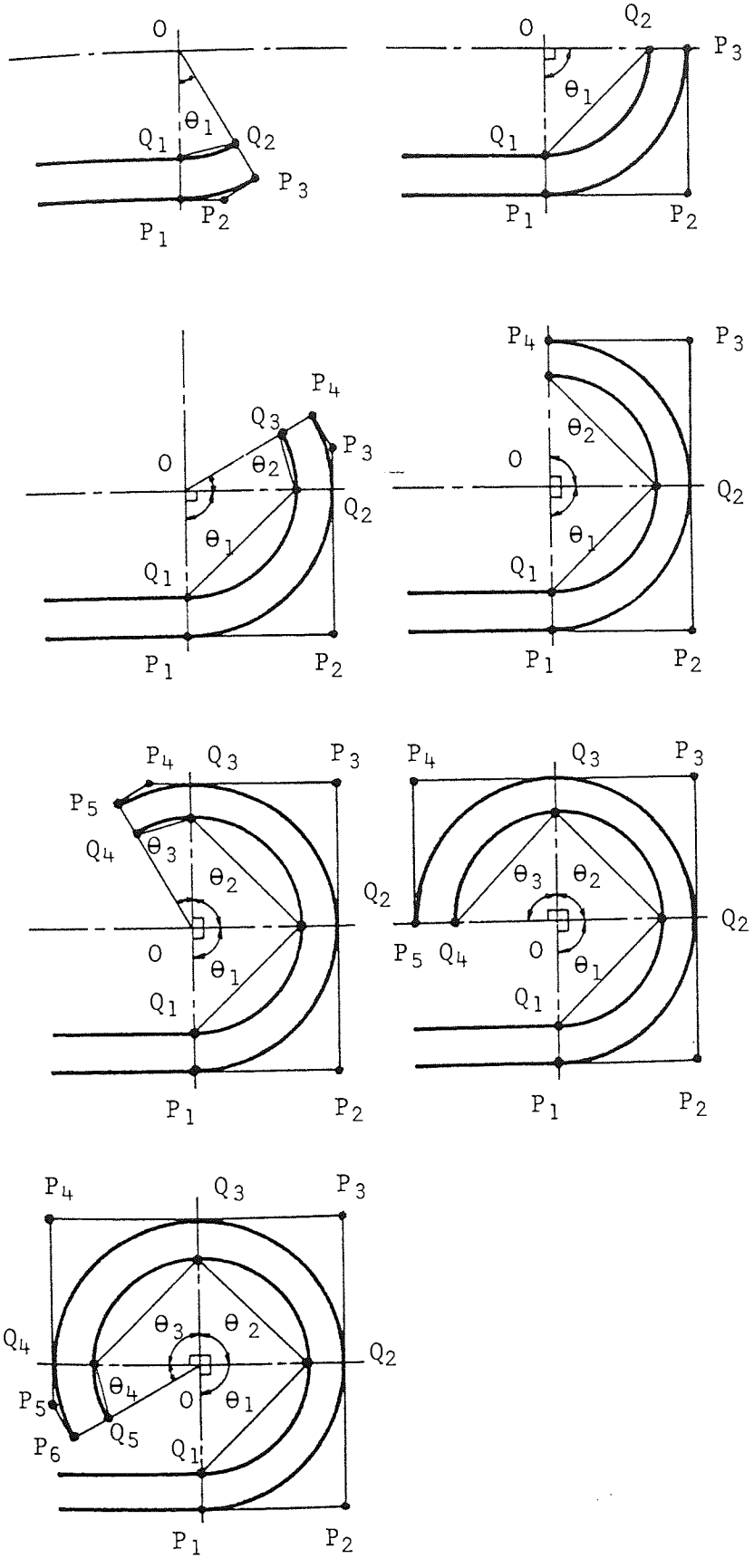
Fig 9.4 ROLL CONTOUR MODIFICATION INVOLVING PART OR WHOLE OF LINEAR AND CIRCULAR ELEMENTS



(a) LINEAR ELEMENT CONTOURS

(b) CIRCULAR ELEMENT CONTOURS

Fig 9.5 DEFINITION OF ROLL CONTOUR ELEMENT ORIENTATION AND POSITION



CASE 1

$|\theta| = \theta_1$

$(< 90^0)$

3 convex points

2 concave points

n = 3

CASE 2

$|\theta| = (\theta_1 + \theta_2)$

$(< 180^0)$

4 convex points

3 concave points

$\theta_1 = 90^0$

n = 4

CASE 3

$|\theta| = (\theta_1 + \theta_2 + \theta_3)$

$(< 270^0)$

5 convex points

3 concave points

$\theta_1 = \theta_2 = 90^0$

n = 5

CASE 4

$|\theta| = (\theta_1+\theta_2+\theta_3+\theta_4)$

6 convex points

5 concave points

$\theta_1 = \theta_2 = \theta_3 = 90^0$

n = 6

Fig 9.7 LINEAR ENVELOPES FOR A CIRCULAR ELEMENT

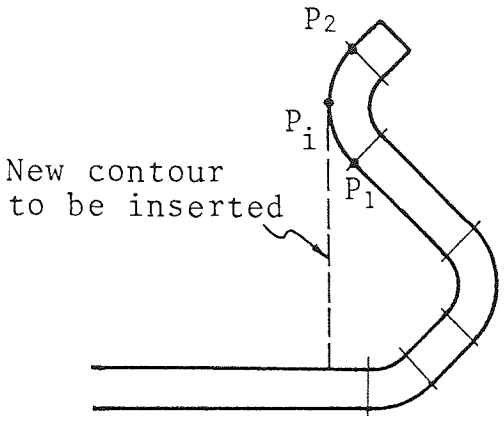


Fig 9.6 A POINT OF UNDEFINED ORIENTATION INVOLVED IN THE MODIFIED ROLL CONTOUR

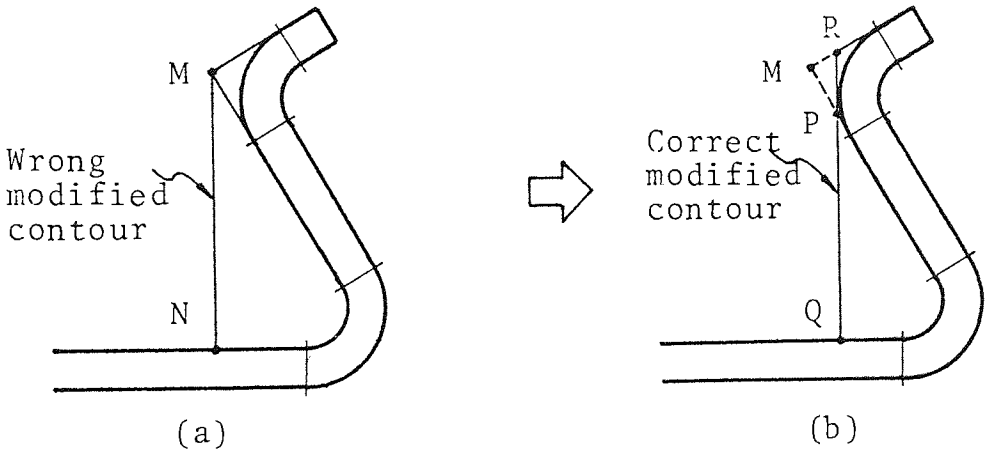


Fig 9.8 THE NECESSITY OF INCLUDING A VERTICAL LINEAR INSERT

Fig 9.9 EFFECT OF INCLUDING THE VERTICAL LINEAR INSERT

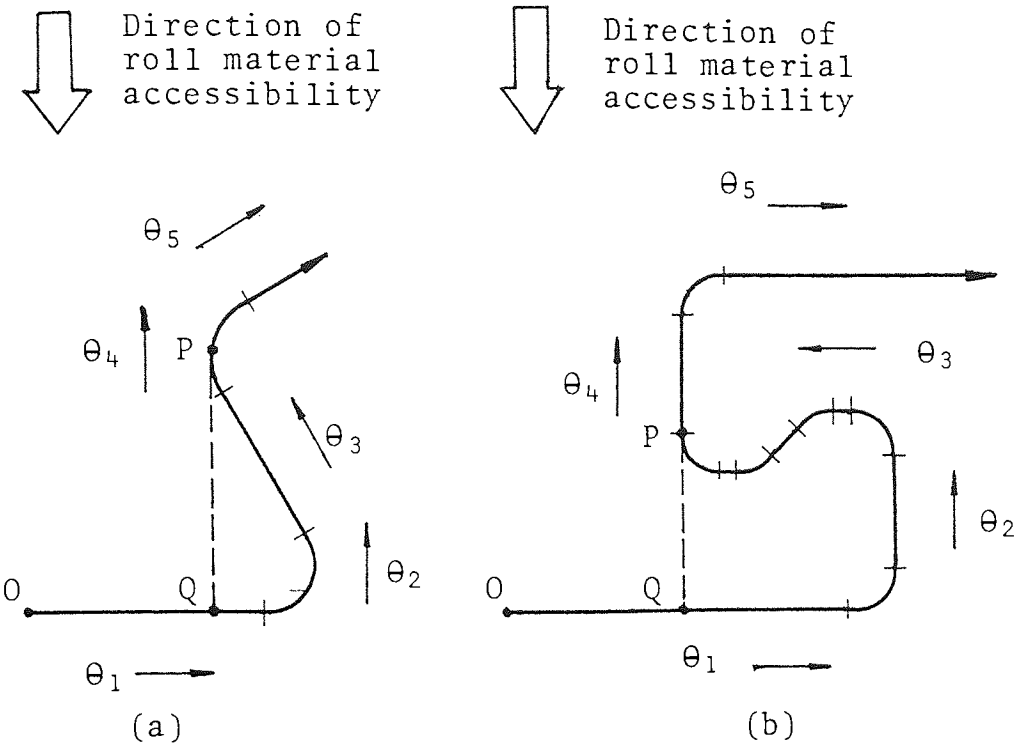


Fig 9.10 BASIC ROLL CONTOUR MODIFICATION FOR THE INITIAL UPWARD BEND CASES

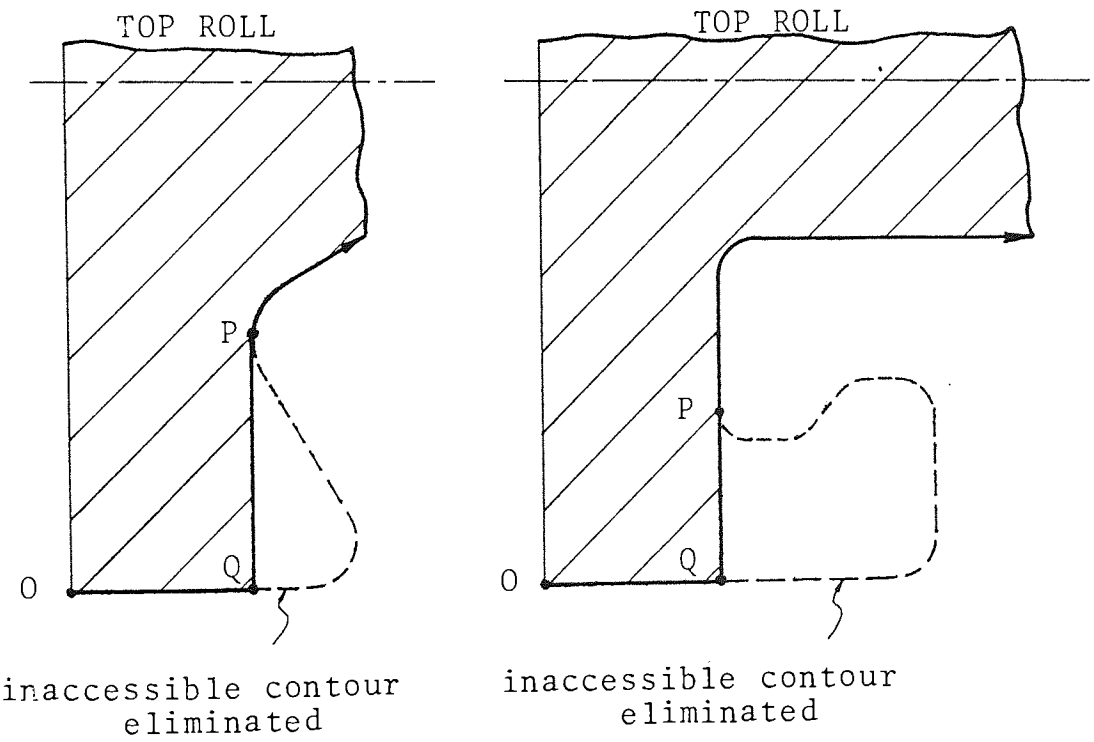


Fig 9.11 RESULTANT ROLL CONTOURS AFTER MODIFICATION IN THE INITIAL UPWARD BEND CASES

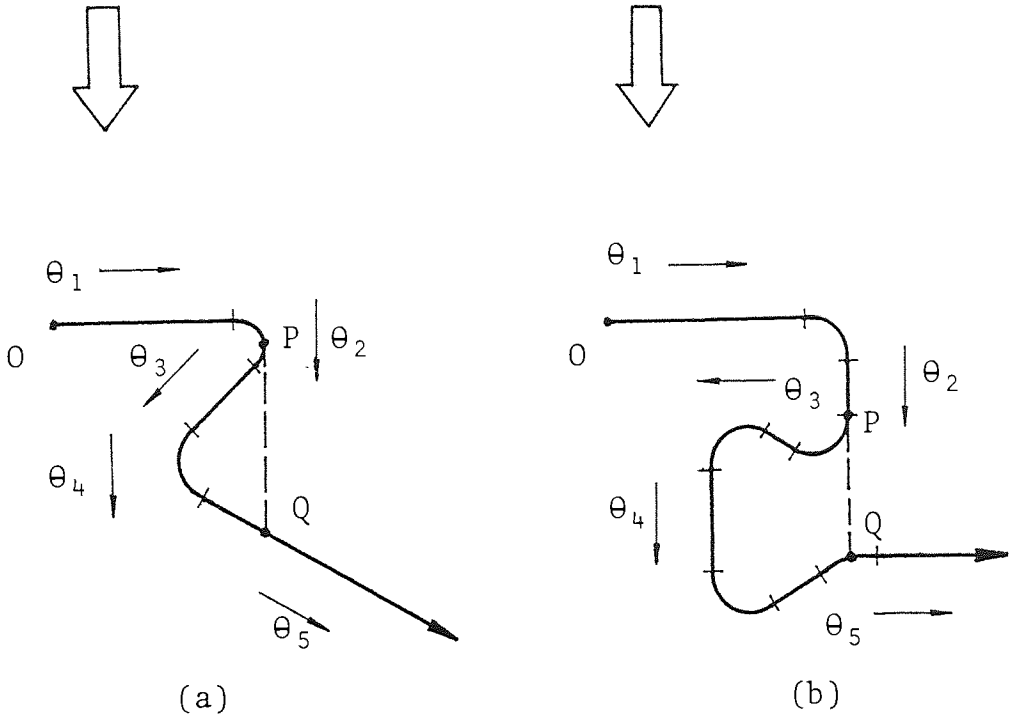


Fig 9.12 BASIC ROLL CONTOUR MODIFICATION FOR THE INITIAL DOWNWARD BEND CASES

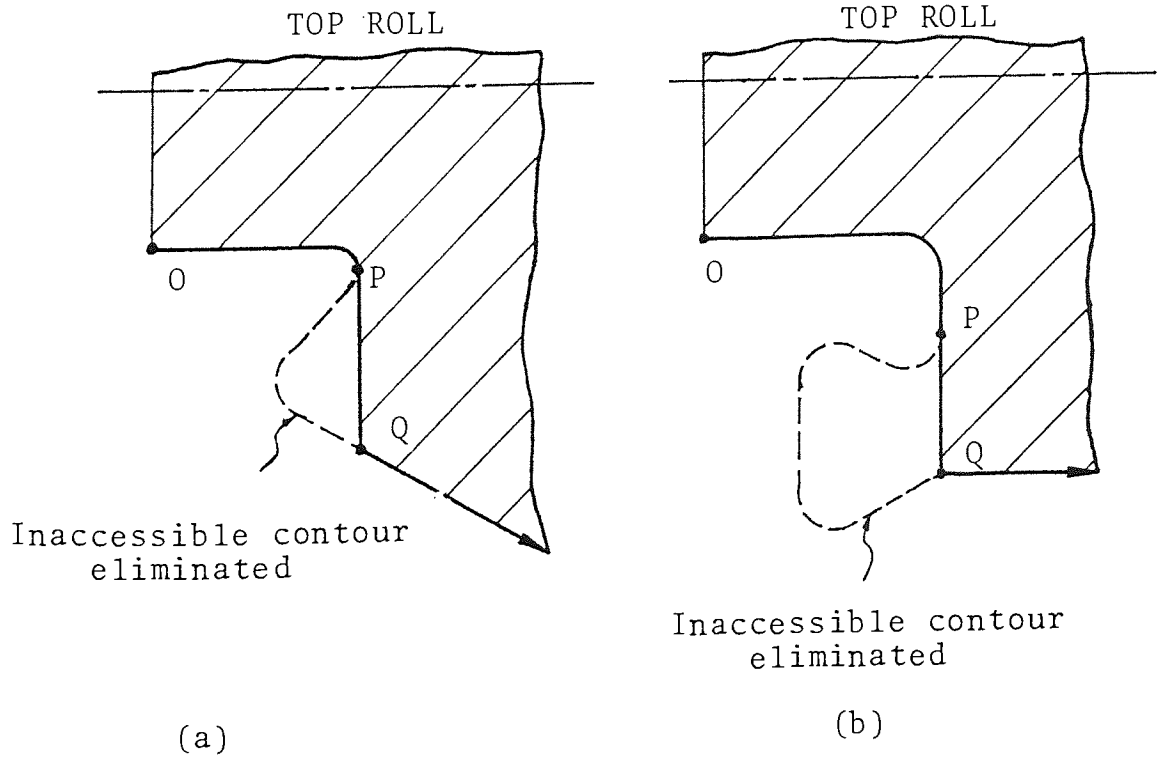


Fig 9.13 RESULTANT ROLL CONTOURS AFTER MODIFICATION IN THE INITIAL DOWNWARD BEND CASES

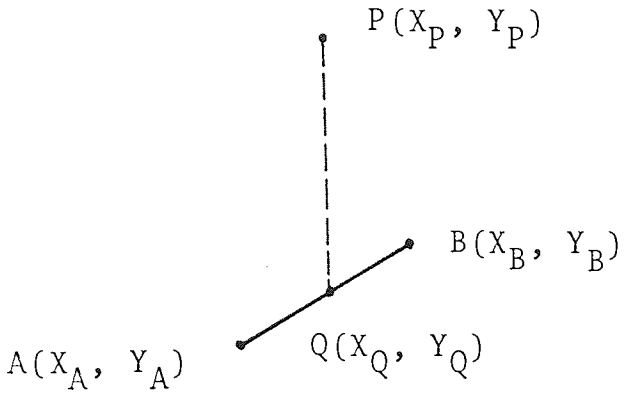
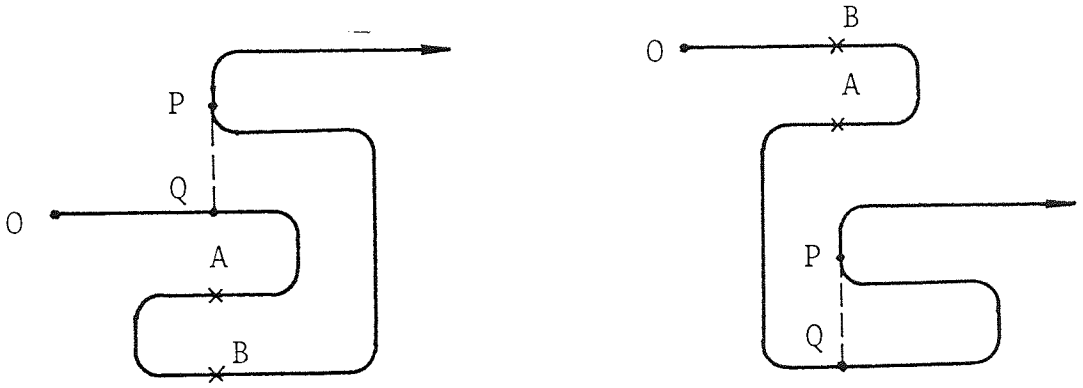
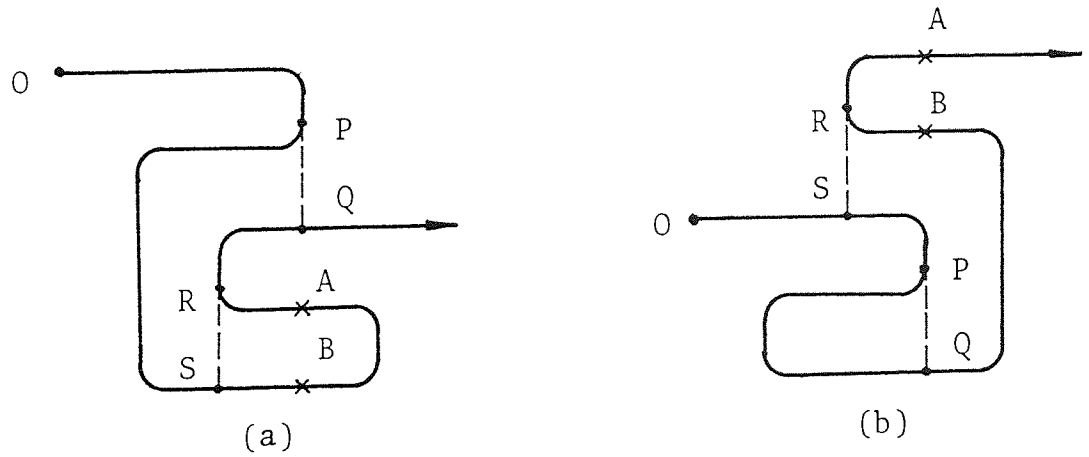


Fig 9.14 COMPUTATION FOR THE MODIFIED ROLL CONTOUR PARTS



$$X_P = X_Q = X_A = X_B$$

Fig 9.15 COMPLICATED TURNS IN THE INITIAL UPWARD BEND CASES



$$X_P = X_Q = X_A = X_B$$

Fig 9.16 COMPLICATED TURNS IN THE INITIAL DOWNWARD BEND CASES

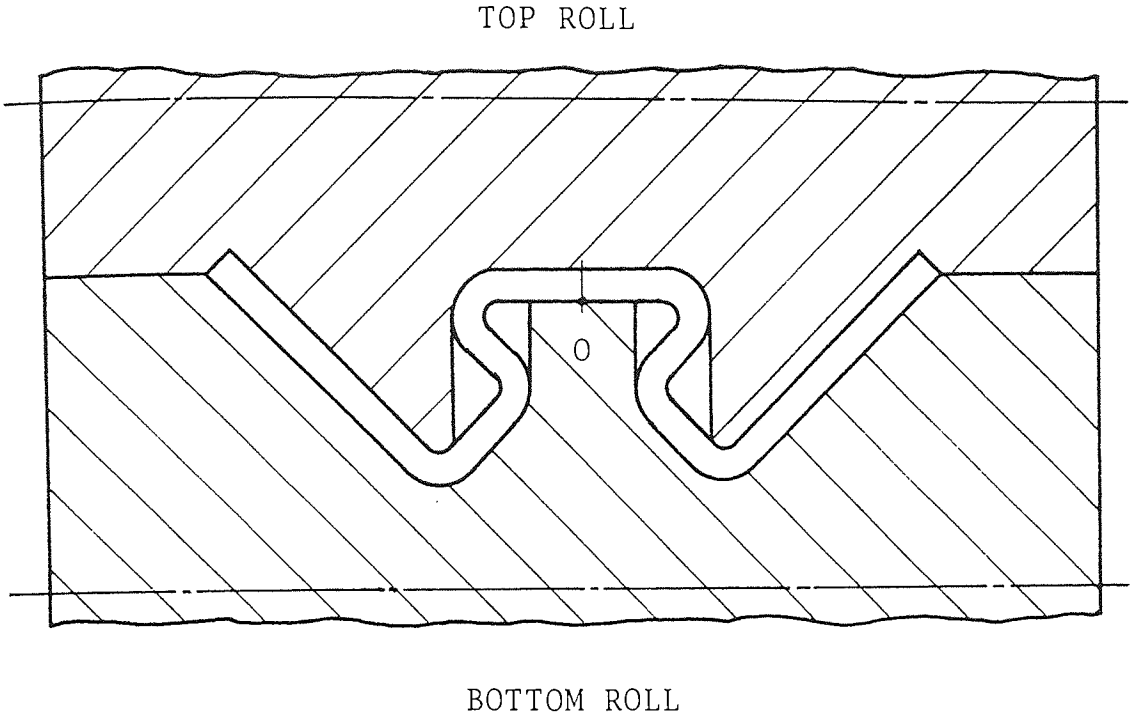


Fig 9.17 CONTOUR MODIFICATIONS FOR TEMPLATE FACES ACCESSIBLE FROM ONLY ONE SIDE

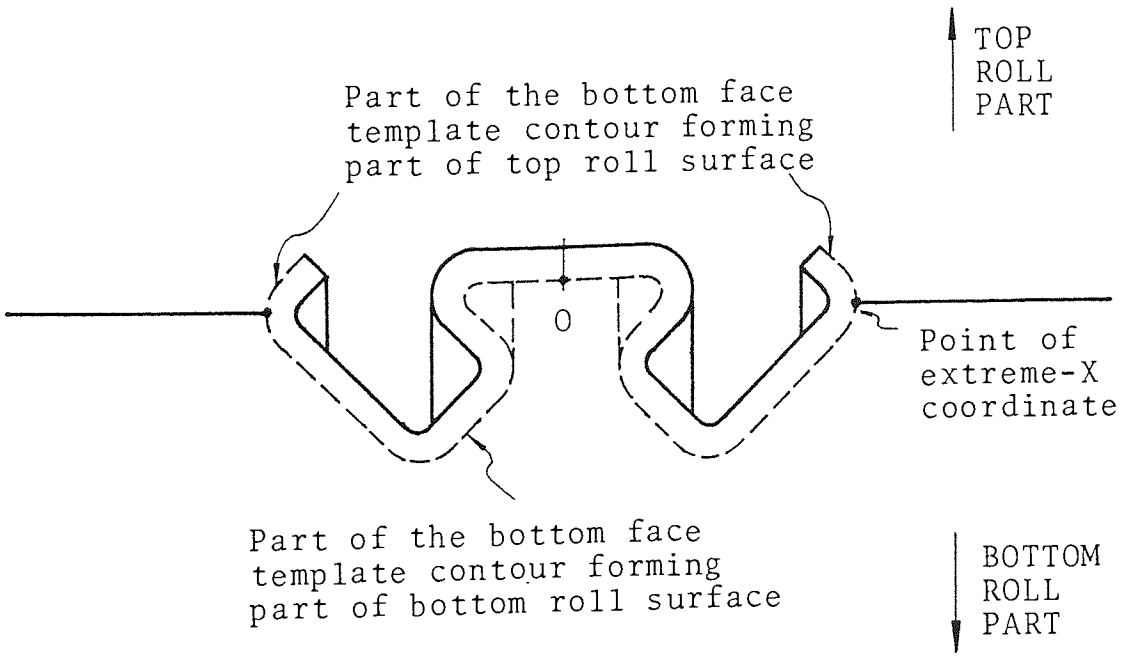


Fig 9.18 SPLITTING OF THE TEMPLATE CONTOUR SURFACE BETWEEN TOP AND BOTTOM ROLLS

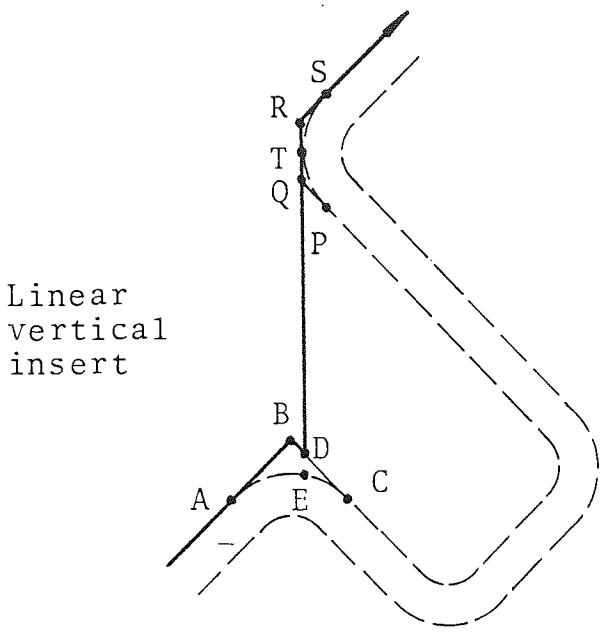


Fig 9.19 RESTORING CIRCULAR ELEMENT CONTOUR IN THE INITIAL UPWARD BEND CASES

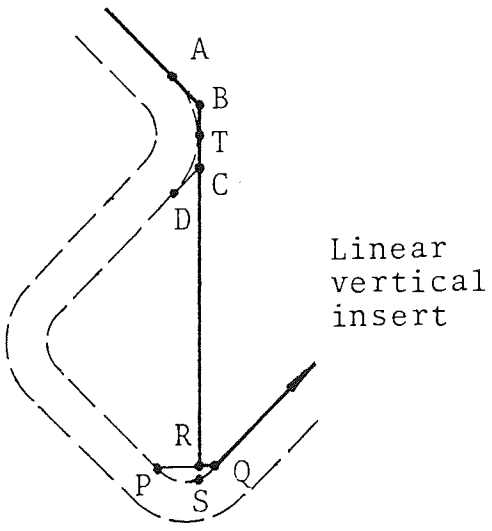
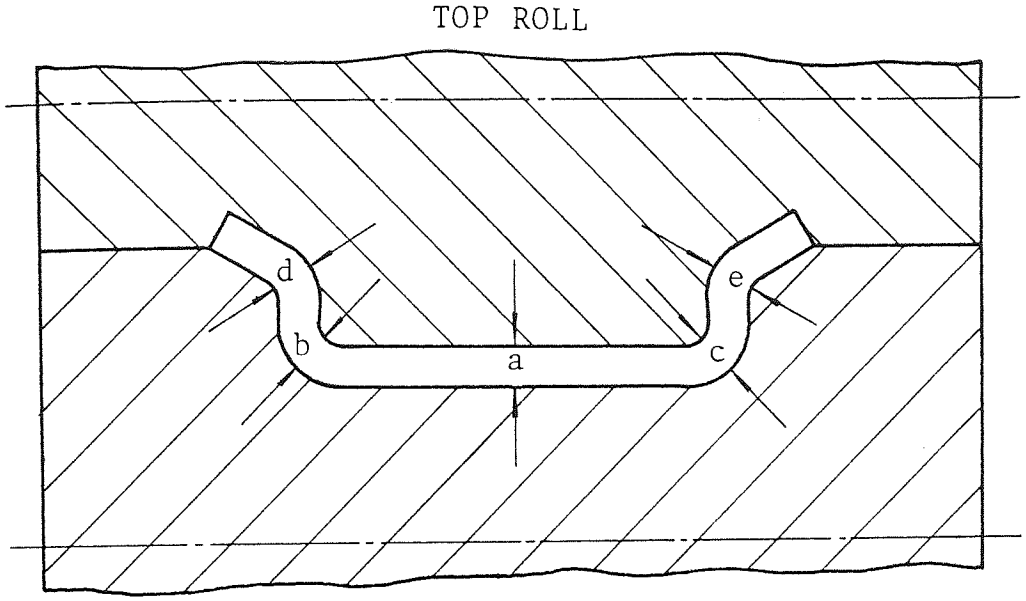


Fig 9.20 RESTORING CIRCULAR ELEMENT CONTOUR IN THE INITIAL DOWNWARD BEND CASES



(Varying gap distances a, b, c, d and e)

Fig 9.21 PINCH - DIFFERENCE SURFACES

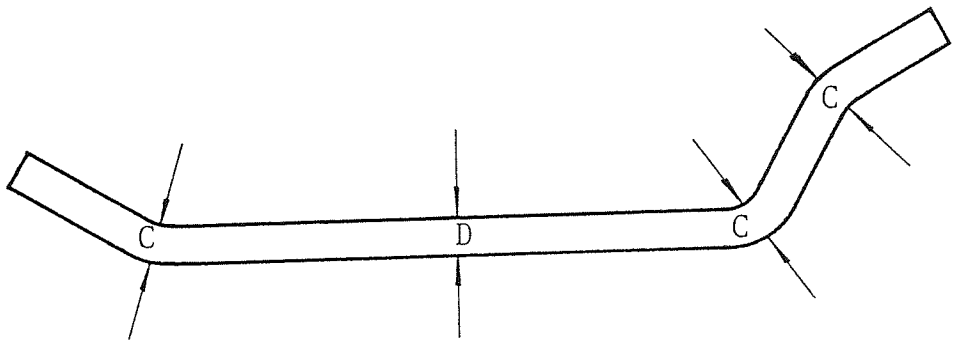


Fig 9.22 THE DOUBLE-THICKNESS METHOD

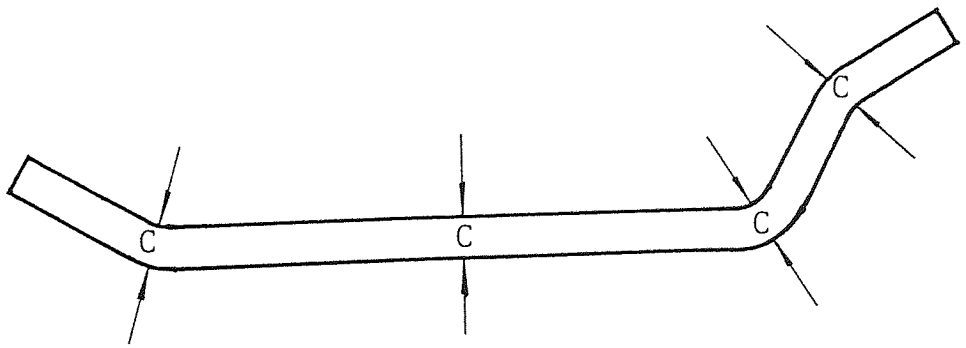


Fig 9.23 THE SINGLE THICKNESS METHOD

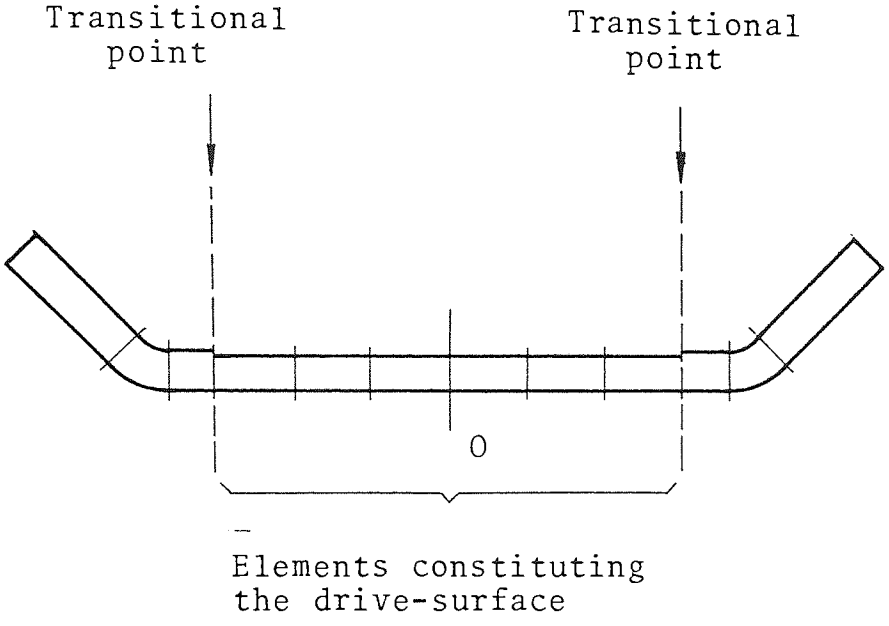
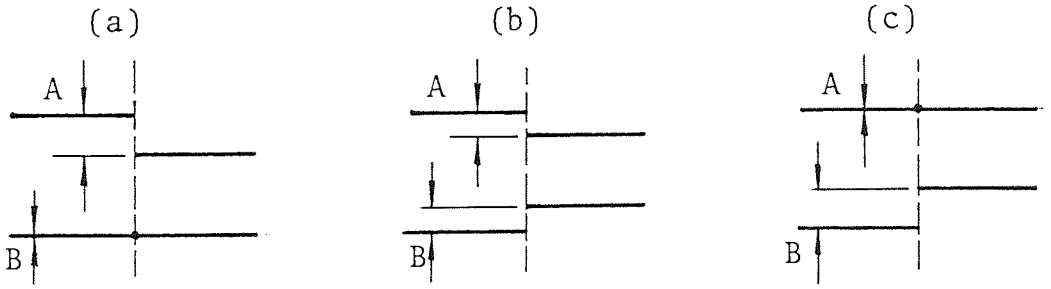


Fig 9.24 THE DRIVE-SURFACE DEFINITION



A ≠ 0
B = 0

A ≠ 0
B ≠ 0

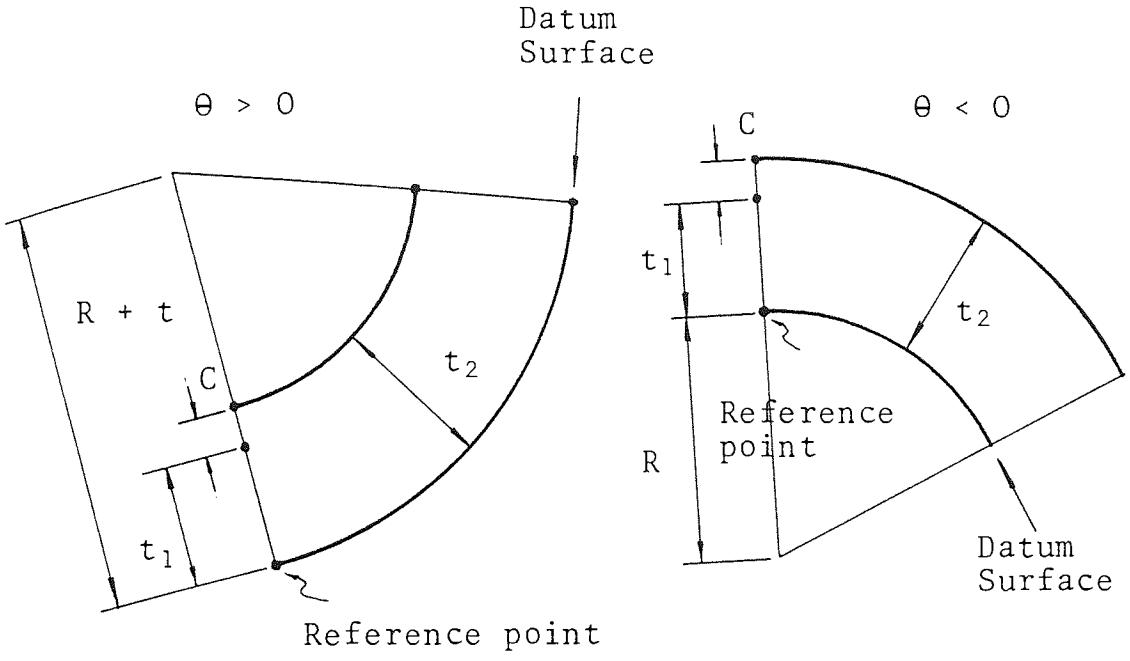
A = 0
B ≠ 0

A = Top face clearance

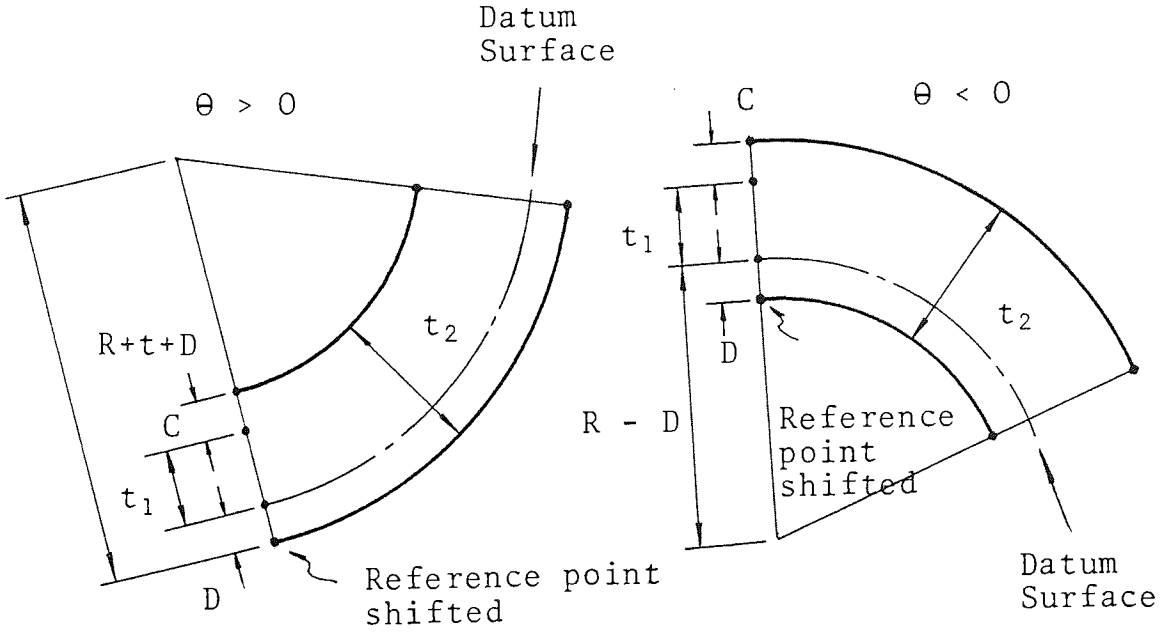
B = Bottom face clearance

Fig 9.25 THREE WAYS OF DISTRIBUTING PINCH-DIFFERENCE CLEARANCE

(t is material thickness, t_1 is drive-surface gap and t_2 is clearance-surface gap)



(a) COMPUTATION WITH UNBROKEN BOTTOM SURFACE



(b) COMPUTATION WITH BROKEN BOTTOM SURFACE

Fig 9.26 COMPUTATION OF PINCH-DIFFERENCE CONTOURS

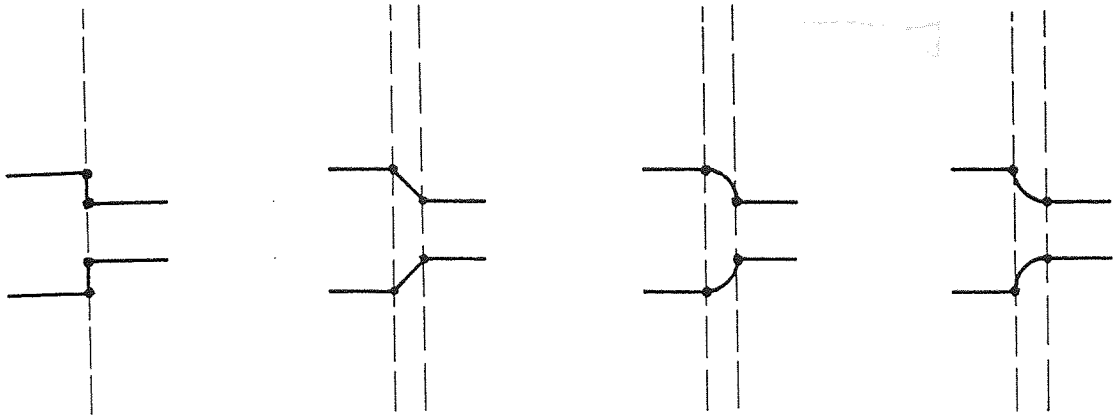


Fig 9.27 ALTERNATIVE METHODS OF ATTACHING THE PINCH-DIFFERENCE SURFACES

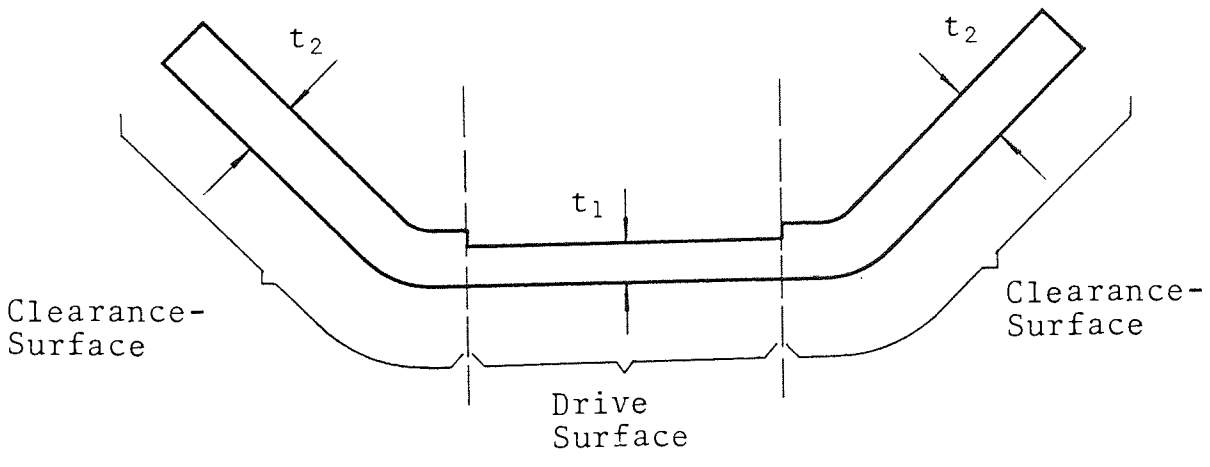


Fig 9.28 THE PINCH-DIFFERENCE SURFACE MODEL

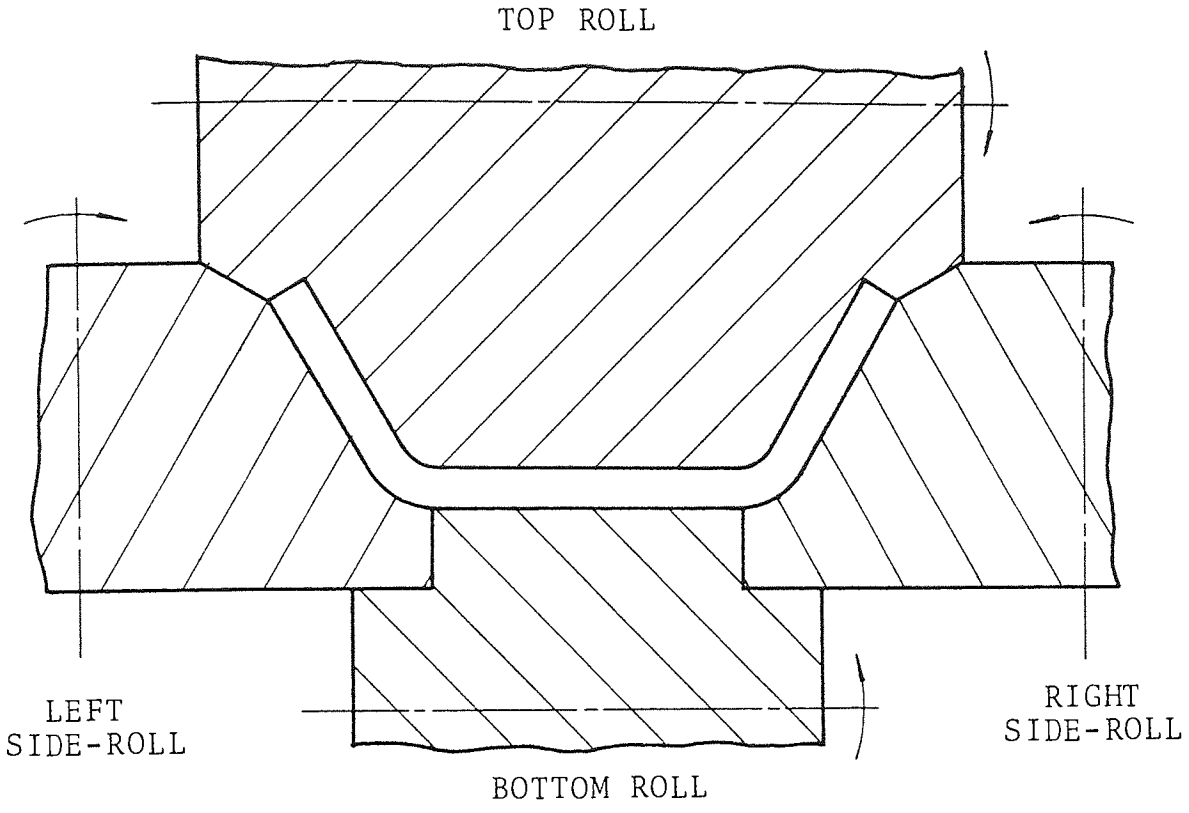
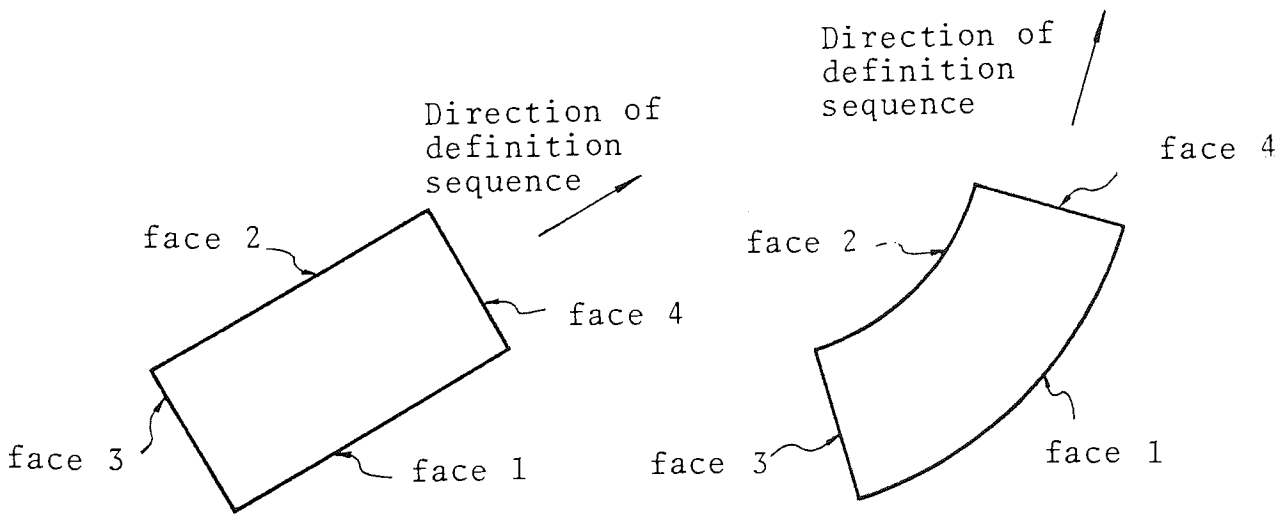


Fig 9.29 THE NECESSITY OF USING SIDE-ROLLS



(a) LINEAR ELEMENT (b) CIRCULAR ELEMENT

Fig 9.30 THE ELEMENT FACES FOR DEFINING SIDE-ROLL CONTOURS

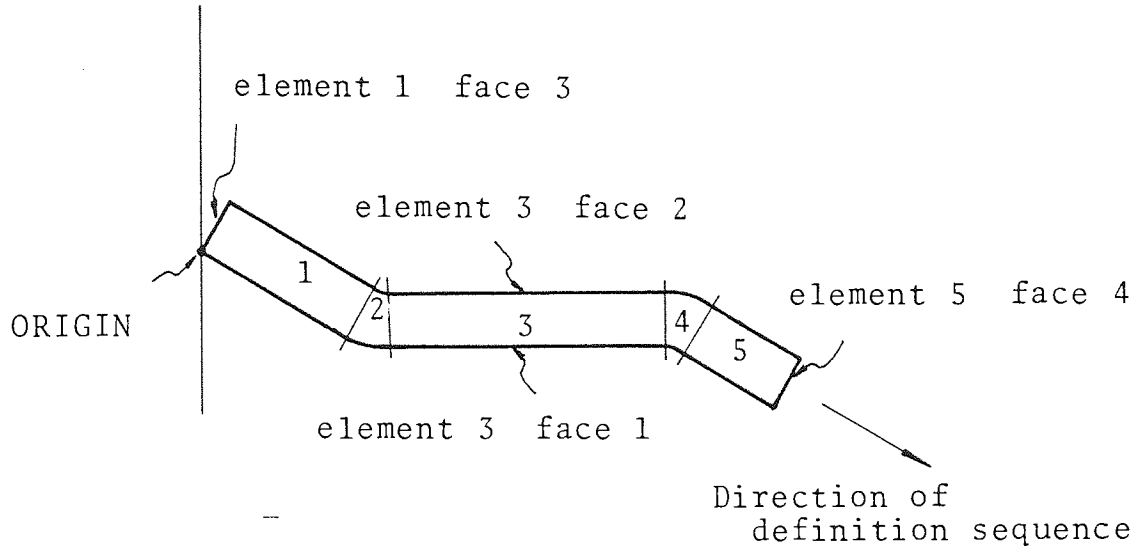


Fig 9.31 OCCURRENCE OF THE ELEMENT FACES

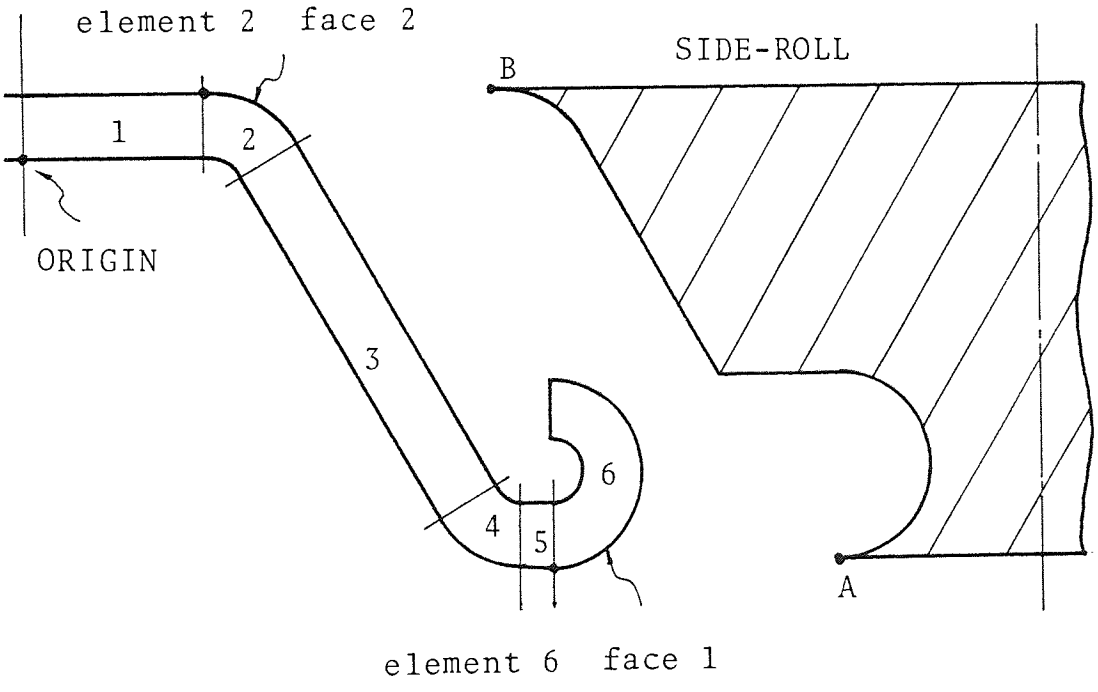


Fig 9.32 AN EXAMPLE OF SIDE-ROLL CONTOUR DEFINITION

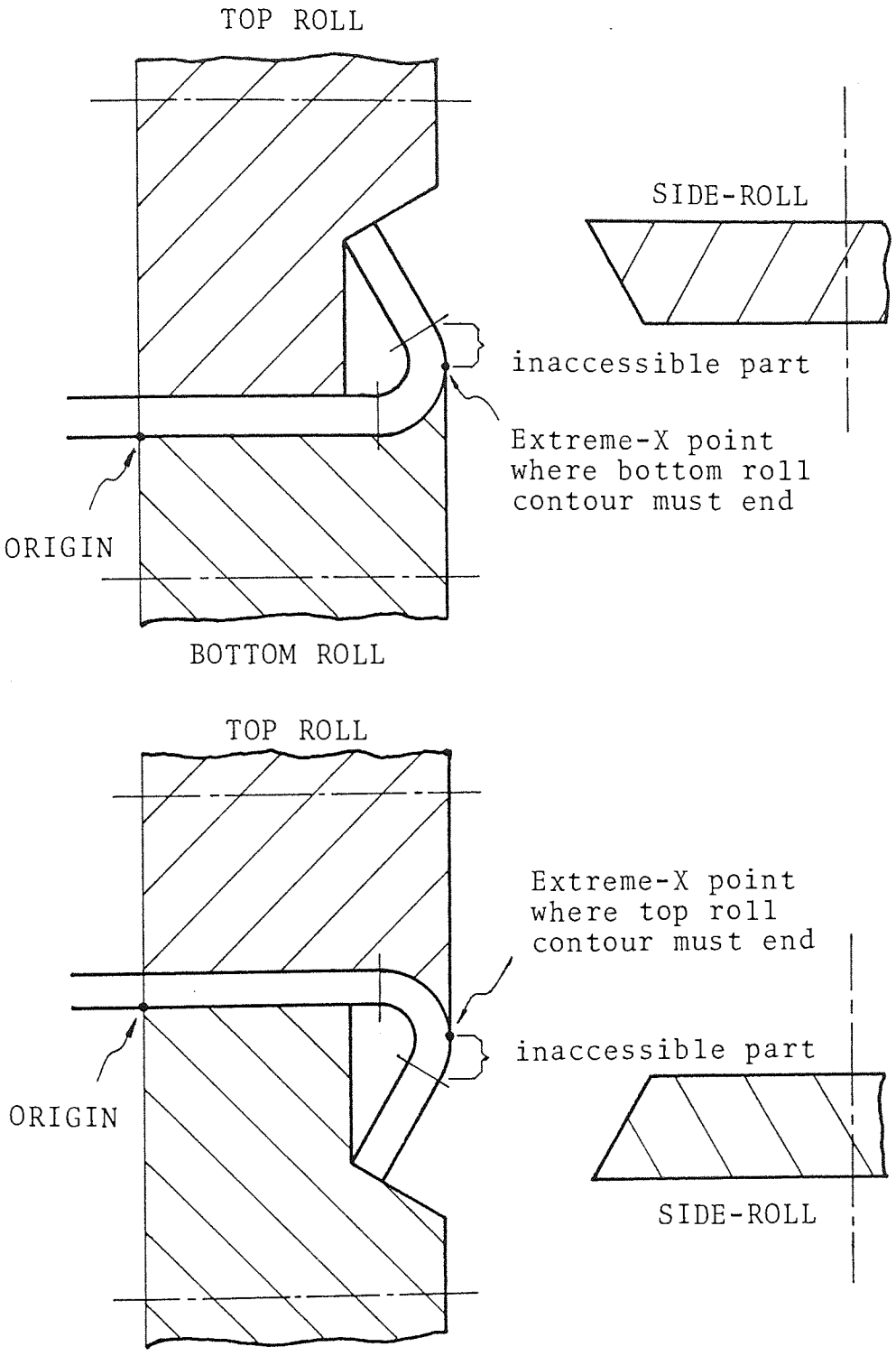


Fig 9.33 CONSEQUENCE OF EXCLUDING THE EXTREME-X POINT IN SIDE-ROLL CONTOUR DEFINITION

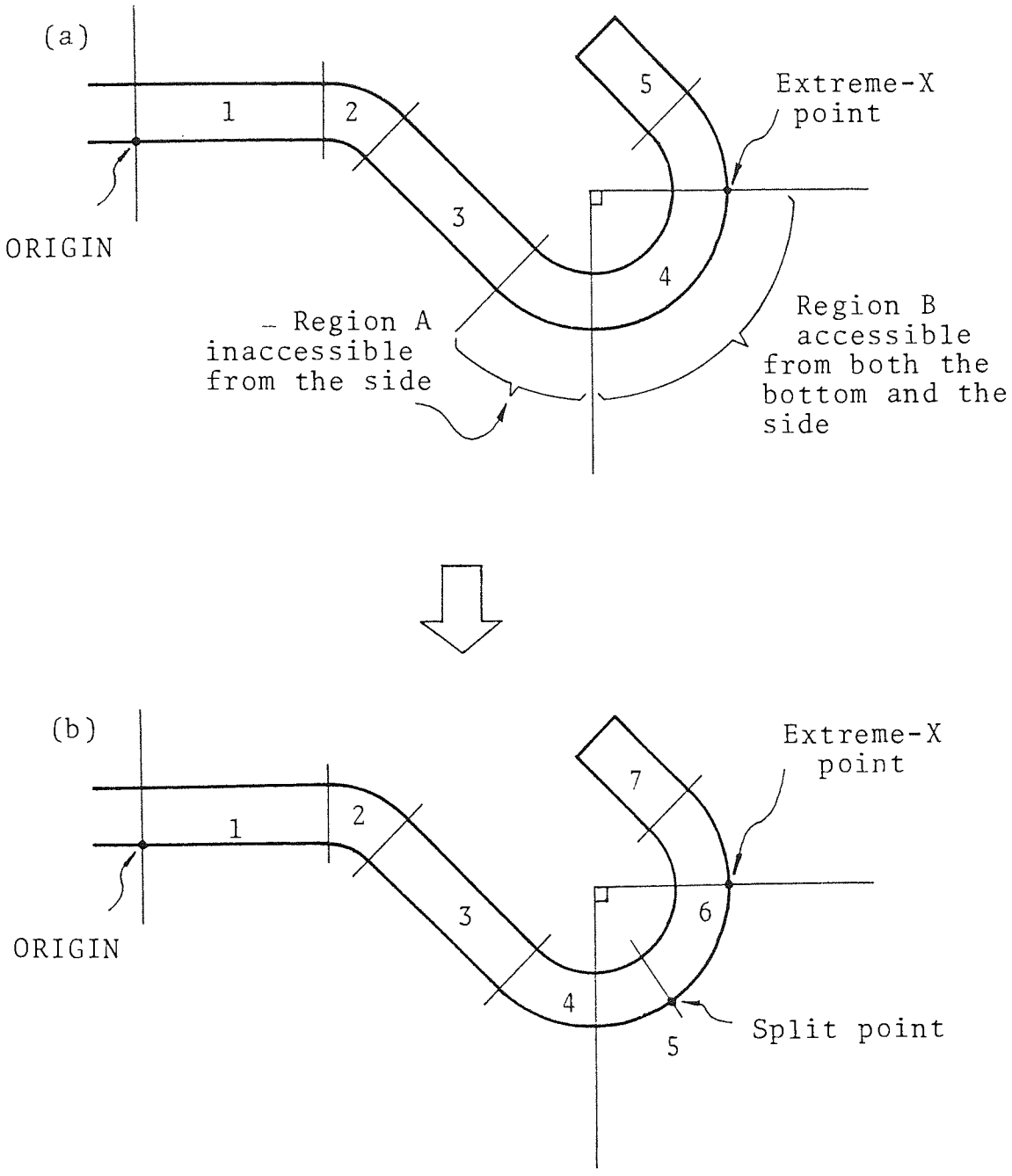


Fig 9.34 SPLITTING THE CIRCULAR ELEMENT FOR SIDE-ROLL CONTOUR DEFINITION

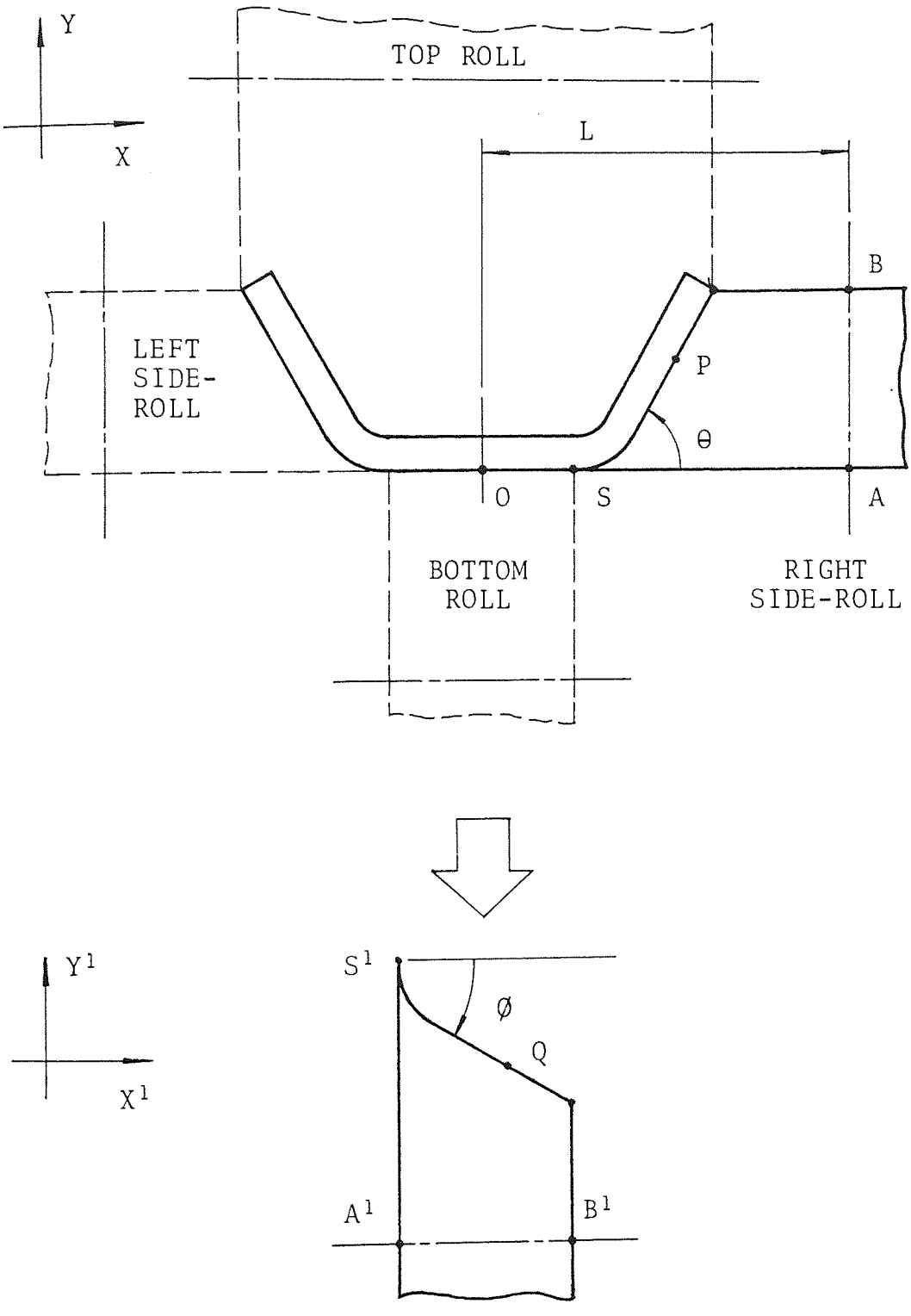
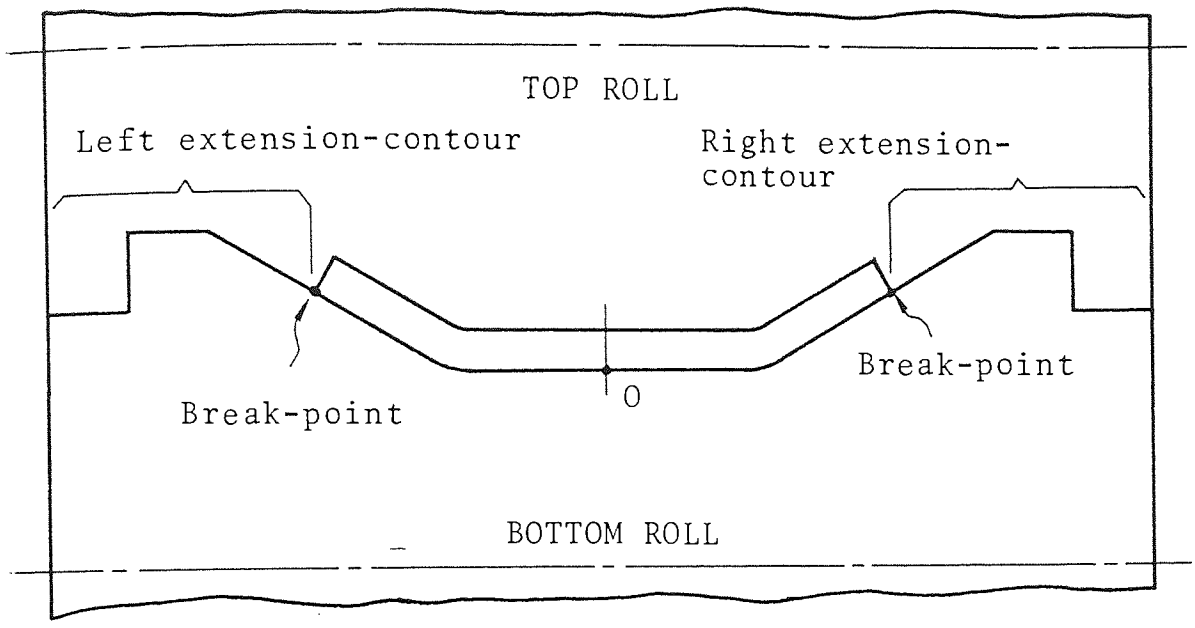
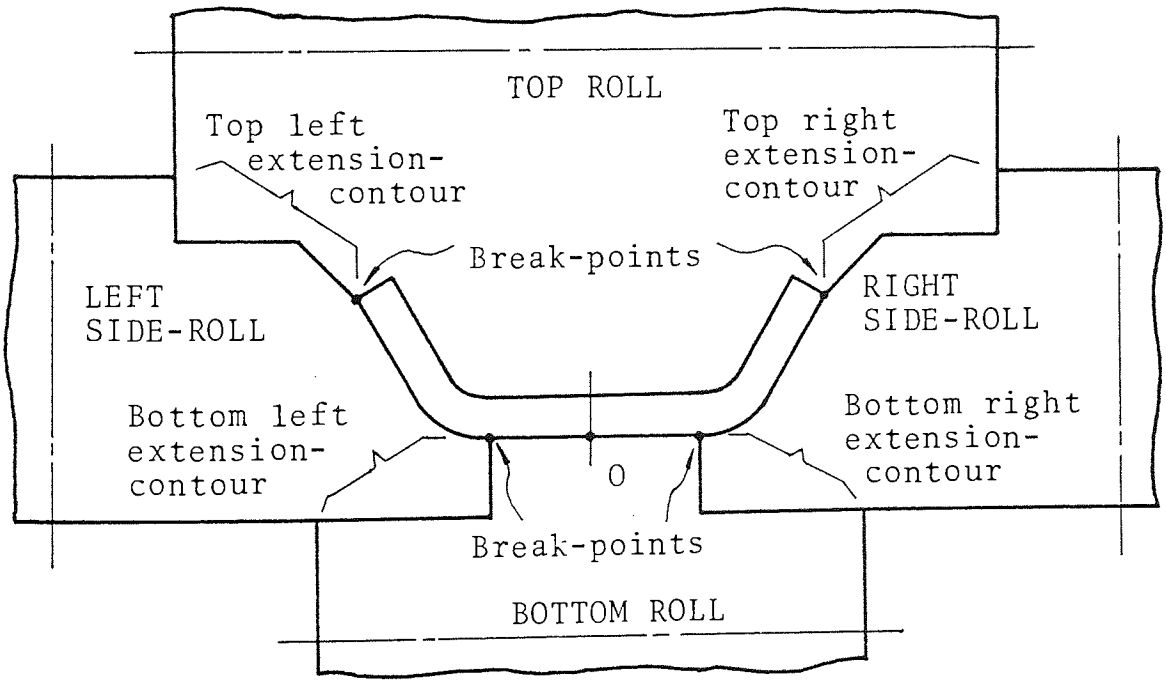


Fig 9.35 CONVERSION FROM TOP AND BOTTOM ROLL ORIENTATION TO SIDE-ROLL ORIENTATION

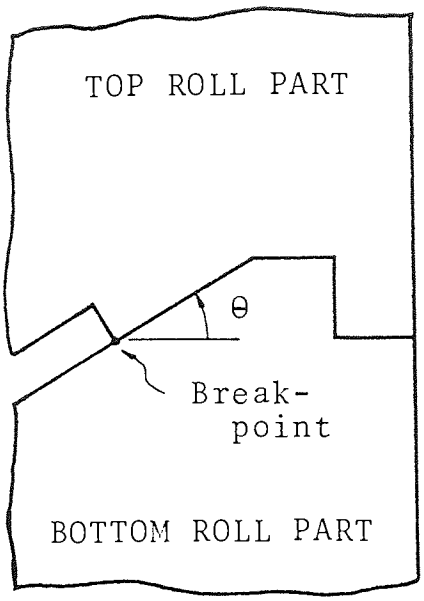


(a) EXTENSION-CONTOURS FOR TOP AND BOTTOM ROLLS ONLY (SPIGOTS)

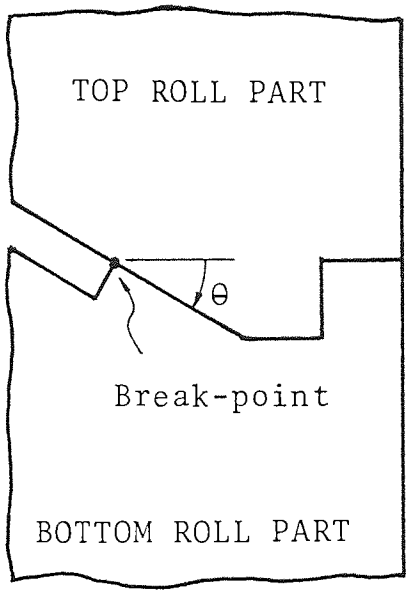


(b) EXTENSION-CONTOURS FOR TOP AND BOTTOM ROLLS WITH SIDE-ROLLS

Fig 9.36 USE OF EXTENSION-CONTOURS FOR PREVENTING SLIDING IN ROLLS

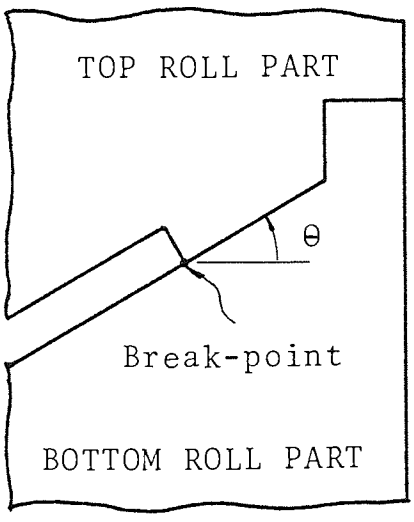


(a) UPWARD BEND

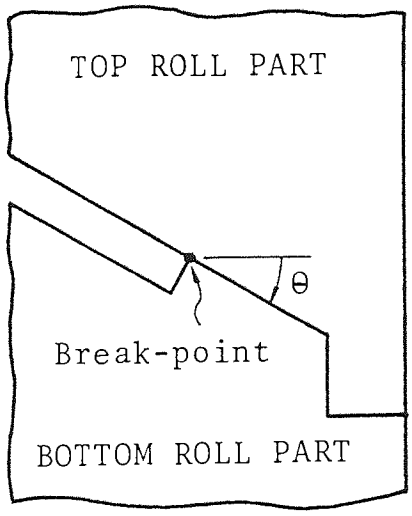


(b) DOWNWARD BEND

Fig 9.37 THE FIRST TYPE OF SPIGOTS

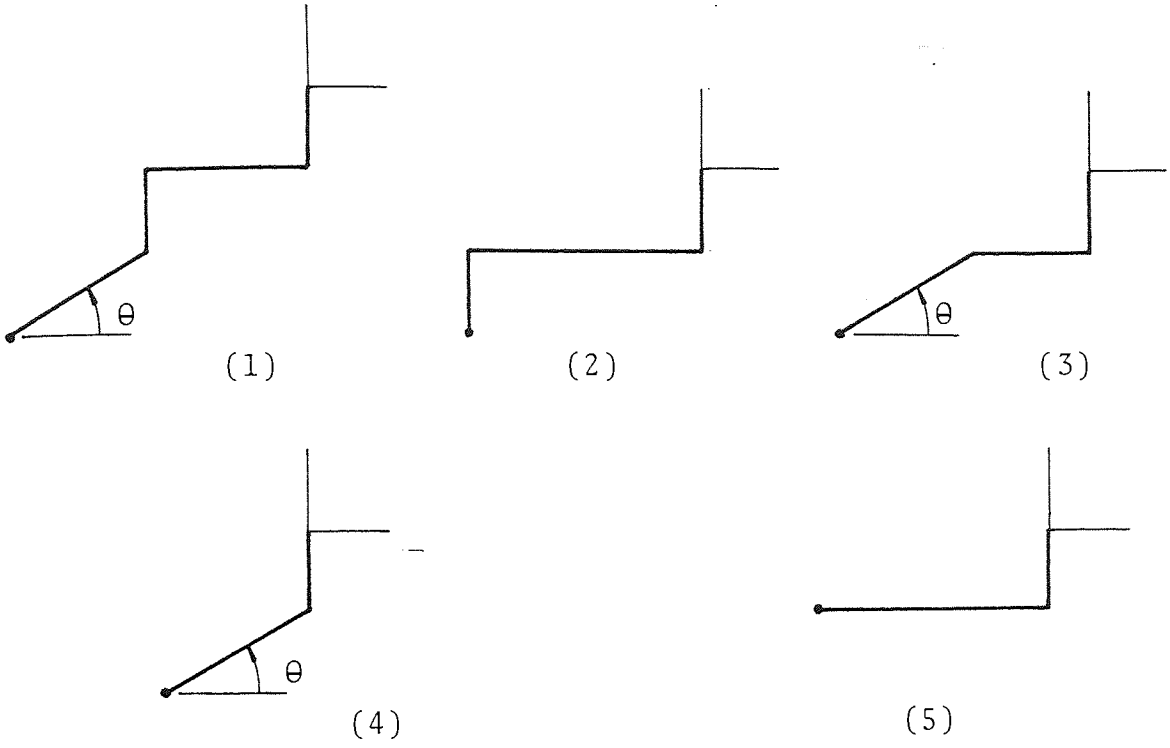


(a) UPWARD BEND

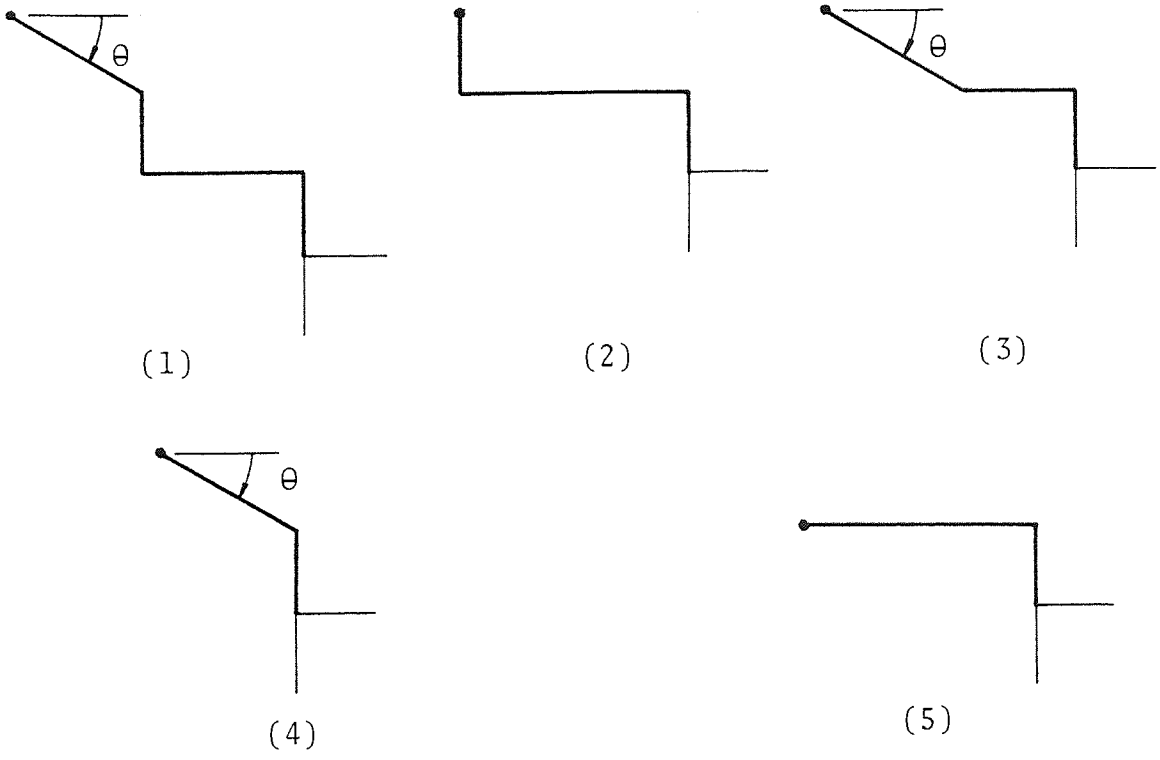


(b) DOWNWARD BEND

Fig 9.38 THE SECOND TYPE OF SPIGOTS

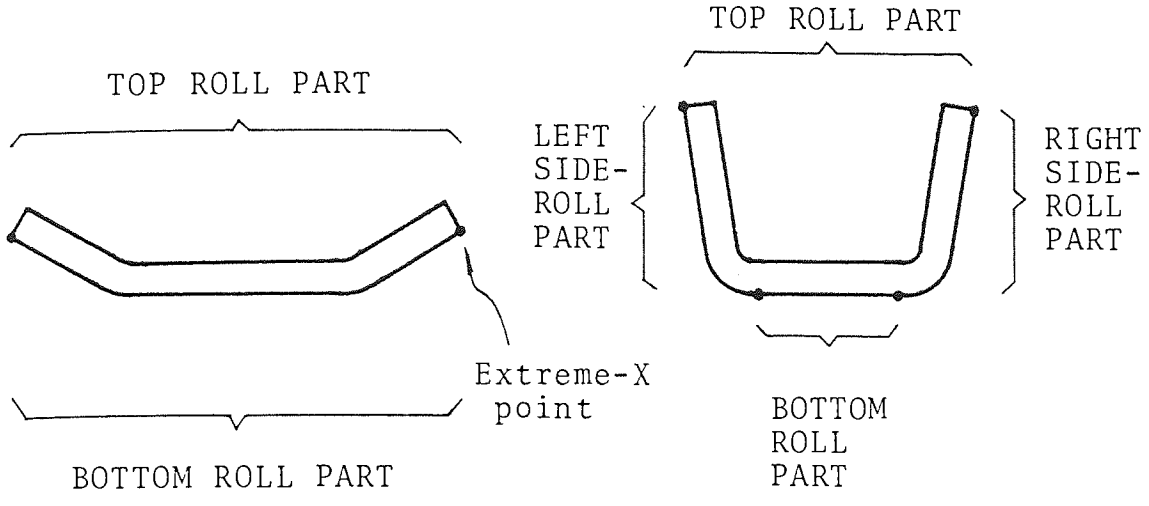


(a) TOP SIDE-ROLL EXTENSION-CONTOURS



(b) BOTTOM SIDE-ROLL EXTENSION-CONTOURS

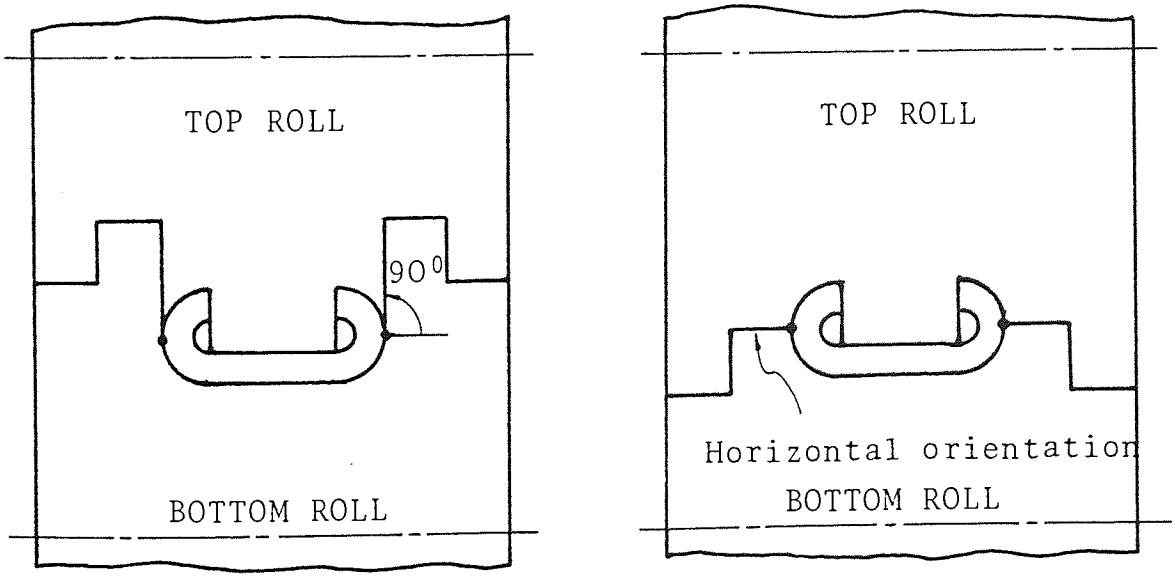
Fig 9.39 TYPES OF SIDE-ROLL EXTENSION-CONTOURS



(a) TWO BREAK-POINTS (WITHOUT SIDE-ROLLS)

(b) FOUR BREAK-POINTS (WITH SIDE-ROLLS)

Fig 9.40 BREAK-POINTS FOR EXTENSION-CONTOURS



(a) UNACCEPTABLE SPIGOT CONTOUR

(b) ACCEPTABLE SPIGOT CONTOUR

Fig 9.41 MONITORING ANGLE OF THE INCLINED PART IN SPIGOTS

TABLE 9.1 CONDITIONS OF VERTICAL ORIENTATION FOR ELEMENTS WITH UPWARD BEND

CONDITION	θ_V	X_V
$\theta_s < -270^o$ and $\theta_e > -270^o$	$-270^o + \theta_n$	$X_c + R$
$\theta_s < -90^o$ and $\theta_e > -90^o$	$-90^o + \theta_n$	$X_c - R$
$\theta_s < 90^o$ and $\theta_e > 90^o$	$90^o + \theta_n$	$X_c + R$
$\theta_s < 270^o$ and $\theta_e > 270^o$	$90^o + \theta_n$	$X_c - R$

TABLE 9.2 CONDITIONS OF VERTICAL ORIENTATION FOR ELEMENTS WITH DOWNWARD BEND

CONDITION	θ_V	X_V
$\theta_s > 270^o$ and $\theta_e < 270^o$	$270^o - \theta_n$	$X_c + R$
$\theta_s > 90^o$ and $\theta_e < 90^o$	$90^o - \theta_n$	$X_c - R$
$\theta_s > -90^o$ and $\theta_e < -90^o$	$-90^o - \theta_n$	$X_c + R$
$\theta_s > -270^o$ and $\theta_e < -270^o$	$-270^o - \theta_n$	$X_c - R$

(where θ_s = Cumulative angle of element orientation at start-point,

θ_e = Cumulative angle of element orientation at end-point,

θ_v = Cumulative angle for vertical insert,

θ_n = Normallising constant to transform θ_s and θ_e to a range between -360^o and $+360^o$,

X_c = X-coordinate of centre-point,

R = Radius of the convex face,

X_v = X-coordinate of the change-points, for vertical insert).

TABLE 9.3 THE ROLL CONTOUR STATUS SUBSTORE

ROXSX(NLMT,12) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION AND VALUE
n,1	Number of linear contours in the element.
n,2	Composite flag for bending direction and radius.
n,3	X-coordinate of centre-point.
n,4	Y-coordinate of centre-point.
n,5	Spare.
n,6	Element type (1 for linear, 2 for circular) and part (+ for concave, - for convex).
n,7	Identifier for pinch-difference elements.
n,8	Identifier for side-roll surface elements.
n,9	Element number for modified point.
n,10	Contour number for modified point.
n,11	Spare.
n,12	Spare.

TABLE 9.4 THE ROLL CONTOUR GEOMETRY SUBSTORE

ROXAX(NLMT,8,4) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION AND VALUE
n,j, 1	X-coordinate of change-point.
n,j, 2	Y-coordinate of change-point.
n,j, 3	Cumulative angle w.r.t. origin.
n,j, 4	Status (1 for linear, 2 for circular and 0 for absence), 0 if j is 1.

(NLMT is the upper-bound of the array size, 100 for basic rolls, 25 for side-rolls. n is the element number and j is the change-point number).

TABLE 9.5 TABLE OF MATERIAL TOLERANCE

THICKNESS OVER	UP TO AND INCLUDING	TOLERANCE
-	0.030"	0.002"
0.030"	0.044"	0.003"
0.044"	0.055"	0.004"
0.055"	0.069"	0.005"
0.069"	0.118"	0.006"

TABLE 9.6 TABLE OF PINCH-DIFFERENCE VALUES

ANGLE OF LEG	PINCH-DIFFERENCE VALUE (IN 0.001") CORRESPONDING TO MAXIMUM THICKNESS OF		
	0.010"	0.006"	0.004"
30^{o}	9	5	$3\frac{1}{2}$
45^{o}	7	4	3
60^{o}	5	3	2
80^{o}	2	1	$\frac{3}{4}$

TABLE 9.7 PINCH-DIFFERENCE INSERT STORE

PLOXX(n,2,4) - REAL NUMBER

SUBSCRIPTS	DESCRIPTION
n,I, 1	X-coordinate of insert end-point.
n,I, 2	Y-coordinate of insert end-point.
n,I, 3	Cumulative angle of inclination.
n,I, 4	Flag indicating presence of insert.

(n is the element number from 1 to 50, I is 1 if the insert is at the start of the element and 2 if it is at the end).

TABLE 9.8 PINCH-DIFFERENCE SURFACE MODE FLAGS

JSURF(3) - INTEGER

SUBSCRIPTS	DESCRIPTION
1	Mode of the previous element.
2	Mode of the present element.
3	Mode of the next element.

(Mode is 1 for drive-surface and 0 for clearance-surface).

TABLE 9.9 PINCH-DIFFERENCE THICKNESS STORE

T(4) - REAL NUMBER

SUBSCRIPT	DESCRIPTION
1	Drive-surface gap.
2	Clearance-surface gap.
3	Bottom face clearance.
4	Top face clearance.

TABLE 9.10 INSERT MODE FLAGS

IDIR(3) - INTEGER

SUBSCRIPT	DESCRIPTION
1	Insert direction at the starting end of element (+1 for upwards and -1 for downwards).
2	Insert direction at the finishing end of element (+1 for upwards and -1 for downwards).
3	Current gap corresponding to the surface mode.

TABLE 9.11 EXTENSION-CONTOUR DEFINITION SCHEME

CONTOUR TYPE	LINEAR PART NUMBER	ANGLE OF ORIENTATION	LENGTH
Spigot Type A	1	As break-point	Positive
	2	0^{o}	Zero or positive
	3	$+90^{o}$	Positive
	4	0^{o}	Positive
Spigot Type B	1	As break-point	Positive
	2	0^{o}	Zero or positive
	3	-90^{o}	Positive
	4	0^{o}	Positive
Top Extension-Contour For Side-Roll	1	$+30^{o}$	Zero or Positive
	2	$+90^{o}$	Zero or Positive
	3	0^{o}	Zero or Positive
	4	$+90^{o}$	Positive
Bottom Extension-Contour For Side-Roll	1	-30^{o}	Zero or Positive
	2	-90^{o}	Zero or Positive
	3	0^{o}	Zero or Positive
	4	-90^{o}	Positive

(Note that a part with zero length means absence of that part).

TABLE 9.12 THE EXTENSION-CONTOUR DATA STORE

EXCONX(4,5,3) - REAL VALUE

SUBSCRIPTS	DESCRIPTION
I,J, 1	X-coordinate of change-point.
I,J, 2	Y-coordinate of change-point.
I,J, 3	Cumulative angle at change-point.

(I is a number indicating the roll contour part, 1 for top roll, 2 for bottom roll, 3 for top-end of side-roll and 4 for bottom-end of side-roll. J is the change-point number, 5 points are required for 4 parts of each extension-contour).

TABLE 9.13 THE BREAK-POINT DATA STORE

BRKPTX(4,3) - REAL VALUE

SUBSCRIPTS	DESCRIPTION
I,1	X-coordinate of break-point.
I,2	Y-coordinate of break-point.
I,3	Angle of inclination at break-point.

(The value of I is as that for Table 9.12).

TABLE 9.14 SUBROUTINES OF THE ROLL DESIGN SOFTWARE (PART 1)

NAME	LEVEL	FUNCTION
RPLOT4	1	Calls the subordinate subroutines to decode data from the flower pattern input file, the roll control input file and the intermediate file, to generate basic roll design contours with pinch-difference surfaces, side-rolls and extension-contours if required and to output the roll contour data for further processing.
RDPL	2	(see Table 7.1).
DCOFR	2	(see the Flower Pattern Input Decoder).
DCORL	2	(see the Roll Input Decoder).
CTEMP	2	Controls the roll contour processing for the total number of bending stages and the update of element bending status at each stage.
STGTM	3	(see Table 8.1).
CLENG	3	(see the Element Length Monitor).
INITP	3	Initialises the roll contour data stores, sets up reference flags and values for side-roll processing, extension-contour processing and pinch-difference surface processing.
PLOT3	3	Controls the pinch-difference processing, the roll contour processing, the drawing of rolls and the output of roll contour data.
STGSR	4	Sets up identifier flags for all side-roll contour elements and also the reference flags for both side-roll processing and extension-contour processing.
STGPN	4	Sets up identifier flags for all drive-surface elements.

TABLE 9.14 (PART 2)

NAME	LEVEL	FUNCTION
PLOT3A	4	Controls the execution of the pinch-difference computation procedures.
PLOT3B	4	Controls the restoration of the true template contour for drawing if required.
ROLL	4	Converts the template contour data to the roll contour data and incorporates the insert contours at transitional points as a result of pinch-difference adjustments.
WRTRD	4	Sets up the current stage pass-height and centre-to-centre distance values.
ROLLA	4	Controls the roll contour modification procedures, the side-roll processing, the break-point processing and the extension-contour processing.
RDOUT1	4	Generates intermediate roll contour data for printing if required.
ROLLC	4	Controls the restoration of circular element contours, the drawing of roll contours, including the side-roll contours, the extension-contours and the template contours if required.
ROLLD	4	Controls the repacking of the roll contour data in a correct sequence for output purposes.
CSURF	5	Initialises and monitors the required conditions for pinch-difference surfaces computation.
PLOT3C	5	Computes the true template contour.
ROLL1	5	Computes and stores roll contour data from the template contour data, with vertical inserts added for the circular elements if necessary.

TABLE 9.14 (PART 3)

NAME	LEVEL	FUNCTION
SMOD	5	Incorporates the insert contours at transitional points of the pinch-difference surfaces into the roll contour.
RSHOW1	5	Prints the content of intermediate roll data stores if necessary.
RPACK	5	Controls the packing of the roll contour data into top and bottom roll contours.
RPICK	5	Packs the end contour at elementlinto the roll contour data store for one-sided sections.
RSA	5	Is the side-roll contour processor.
CBRK	5	Is the break-point processor.
EXTND	5	Is the extension-contour processor.
PLT1B	5	Draws the template contour.
PLOT6	5	Plots the top and bottom roll contours.
OPTBR	5	Outputs the top or bottom roll contour data, with extension-contours if required.
OPSR	5	Outputs the side-roll contour data, with extension-contours if required.
STL2	6	Computes the linear element geometry with pinch-difference adjustments.
ARC2	6	Computes the circular element geometry with pinch-difference adjustments.

TABLE 9.14 (PART 4)

NAME	LEVEL	FUNCTION
RSHOW	6	Prints the specified roll contour array.
EXTRX	6	Sets up the extreme-X point.
POSTD	6	(see Table 6.3).
STL1	6	(see Table 6.3).
ARC1	6	(see Table 6.3).
ROEND	6	Repacks the roll contour elements into actual top and bottom roll parts separated by the extreme-X point.
RSANG	6	Repacks the relevant part of the top or bottom roll contour in preparation for side-roll contour processing.
RS1	6	Forms the side-roll contour from relevant top and bottom roll parts.
PLOT6A	6	Plots the side-roll contour.
SBRK	6	Computes the break-points for either spigot or side-roll selection.
WBRK	6	Prints the break-point details if required.
COMEX	6	Computes the extension-contour change-points.
PEXT1	6	Plots the spigot contours.
PEXT2	6	Plots the extension-contours for top and bottom rolls with side-roll.

TABLE 9.14 (PART 5)

NAME	LEVEL	FUNCTION
ROLL2	6	Performs roll contour modification by identifying and removing the inaccessible parts when scanning through the roll contour data stores.
GENCIR	6	Restores circular element contours from their linear enveloping form.
RDEXR	6	Outputs the extension-contour data for top or bottom roll.
RDROL	6	Outputs the top or bottom roll contour data.
RDEXS	6	Outputs the extension-contour data for side-roll.
RDSID	6	Outputs the side-roll contour data.
STOMAX	7	Stores the extreme-X point details.
RS2	7	Transforms the angles and coordinates of top and bottom roll orientation to side-roll orientation.
OPRD	7	Outputs data for one roll element contour.
CIRMOD	8	Incorporates vertical inserts for circular element contours.

TABLE 9.15 SUBROUTINES OF THE ROLL INPUT DECODER

NAME	LEVEL	FUNCTION
DCORL	1	Decodes general roll design control and definition data and controls other decoding subroutines.
DPINCH	2	Pinch-difference input data decoder.
DSCON	2	Side-roll and extension-contour input data decoder.

CHART 9.1 DATA FLOW OF THE BASIC ROLL CONTOUR PROCESSING

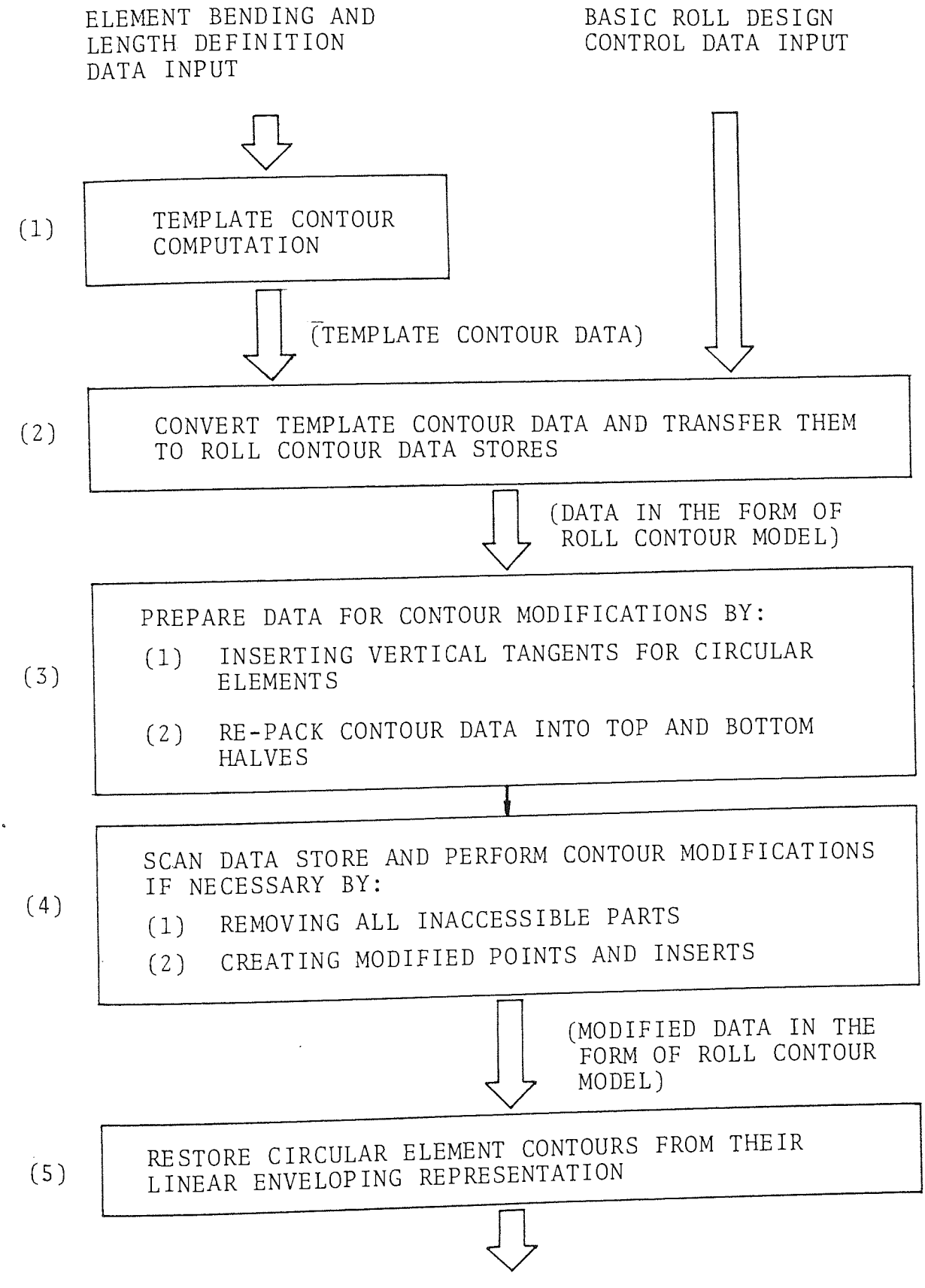


CHART 9.2 DATA FLOW OF THE ROLL CONTOUR PROCESSING INCORPORATING PINCH-DIFFERENCE COMPUTATION

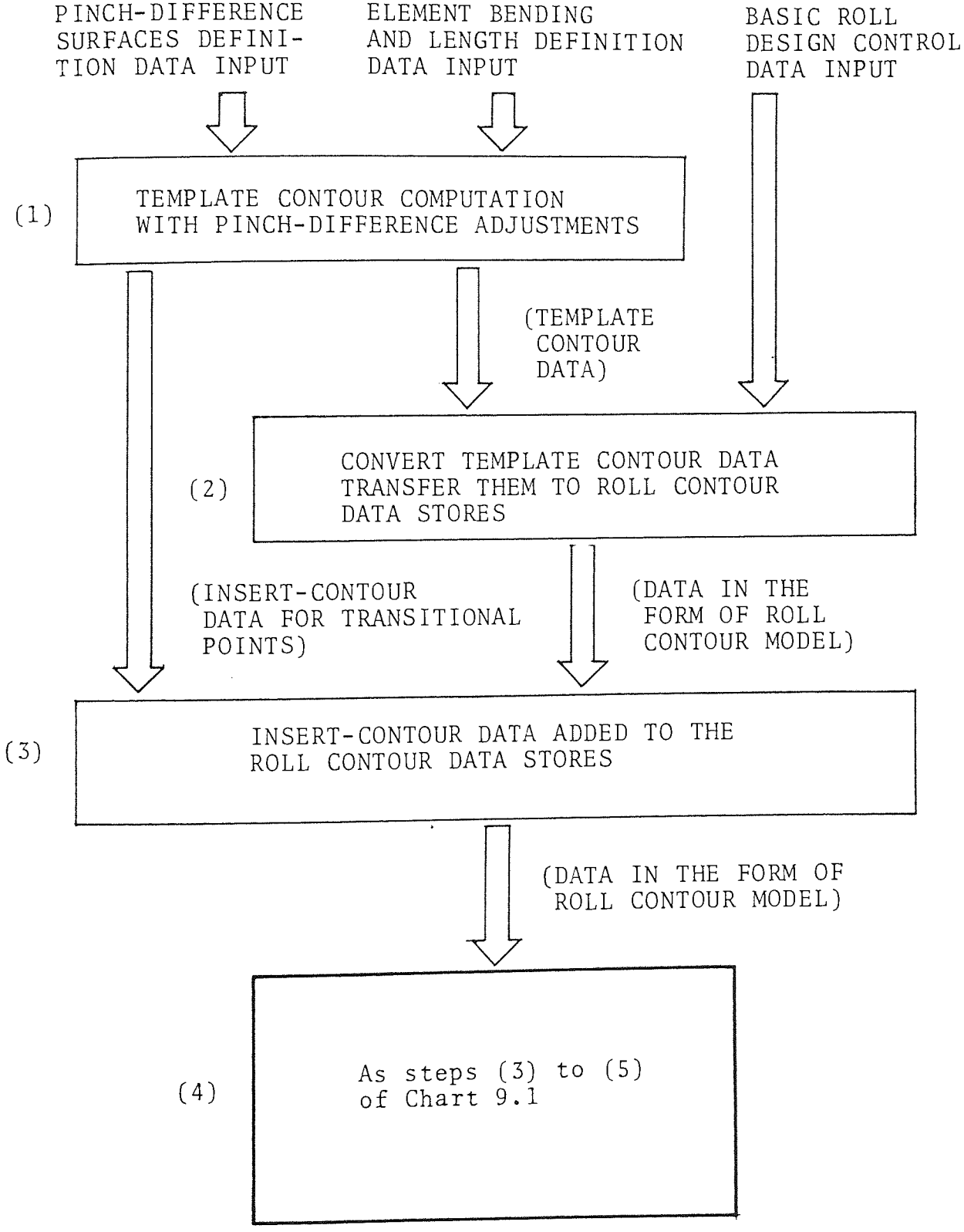


CHART 9.3 DATA FLOW OF THE ROLL CONTOUR PROCESSING INCORPORATING SIDE-ROLLS

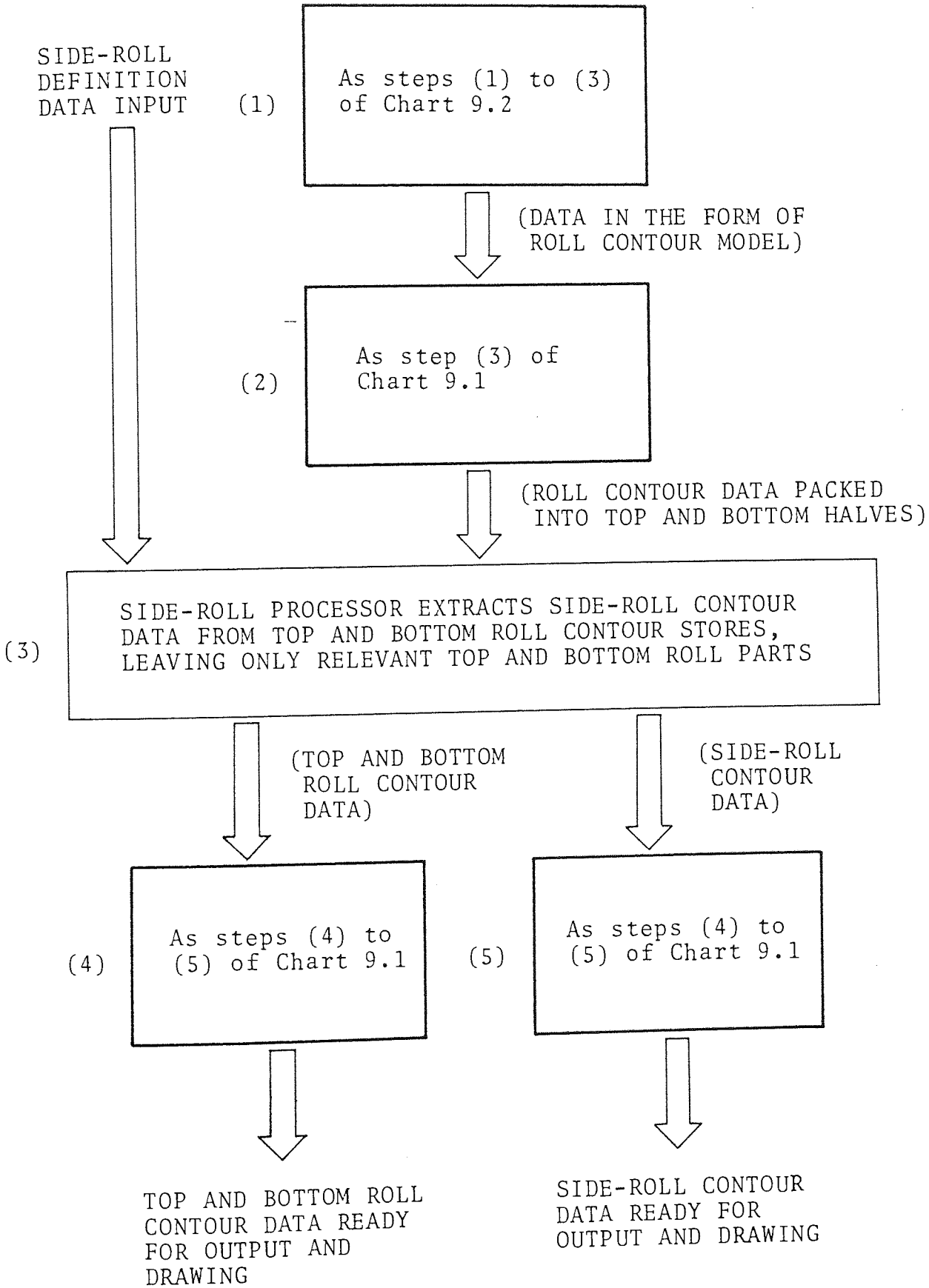


CHART 9.4 DATA FLOW OF THE ROLL CONTOUR PROCESSING INCORPORATING EXTENSION-CONTOURS

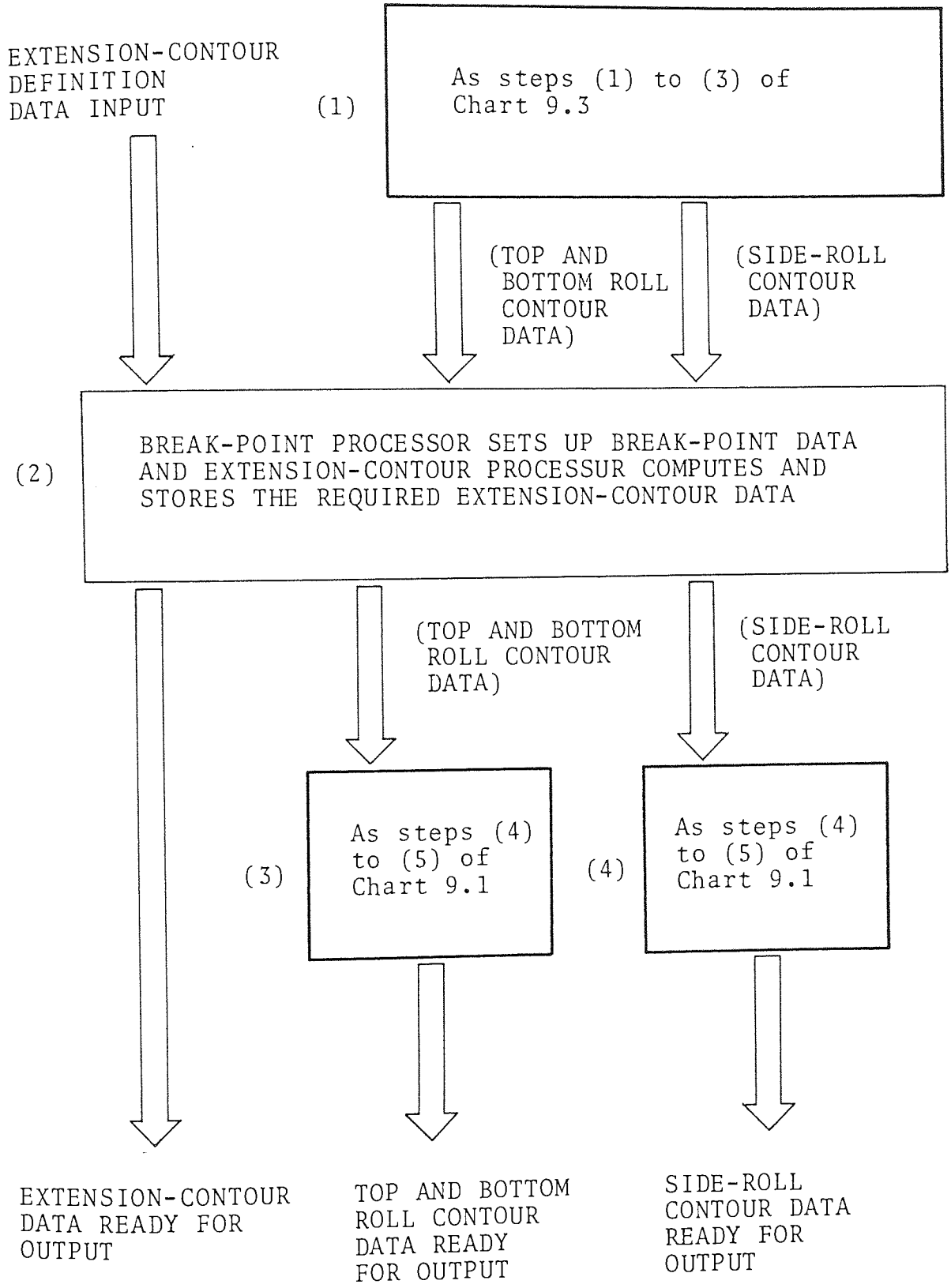


CHART 9.5 HIERARCHY OF THE ROLL DESIGN PROGRAM (RPLOT4)

(PART 1)

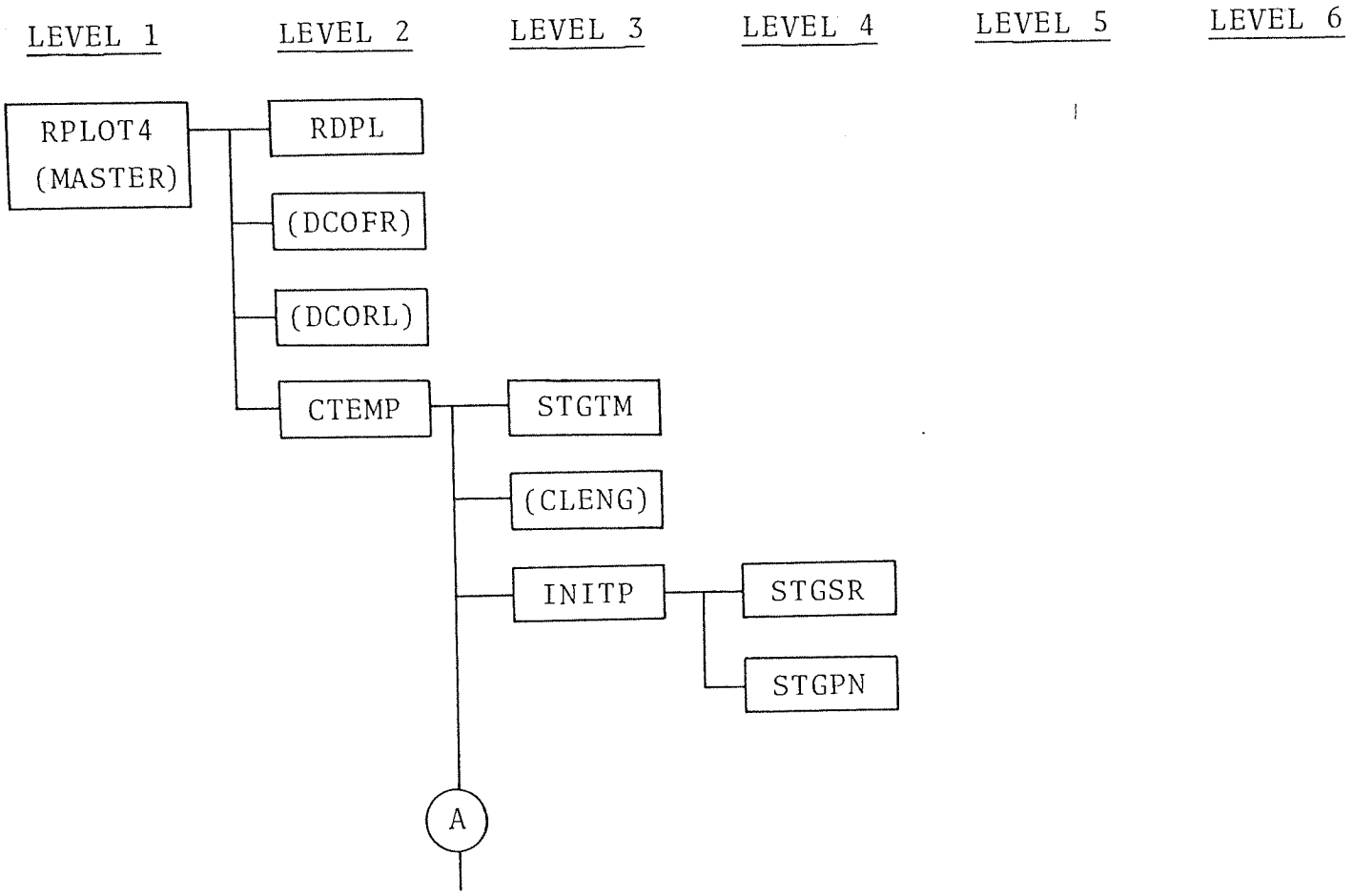


CHART 9.5 (PART 2)

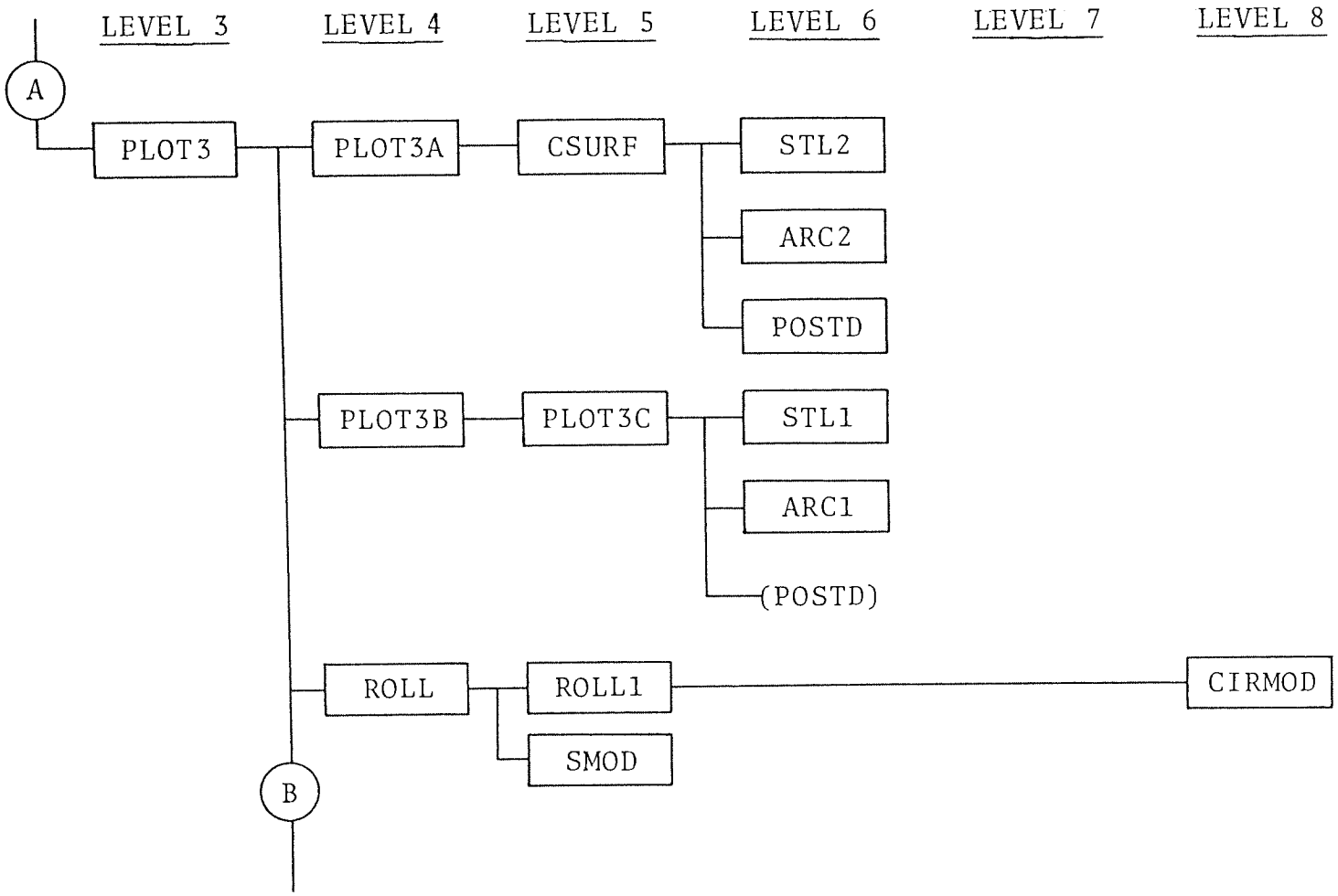


CHART 9.5 (PART 3)

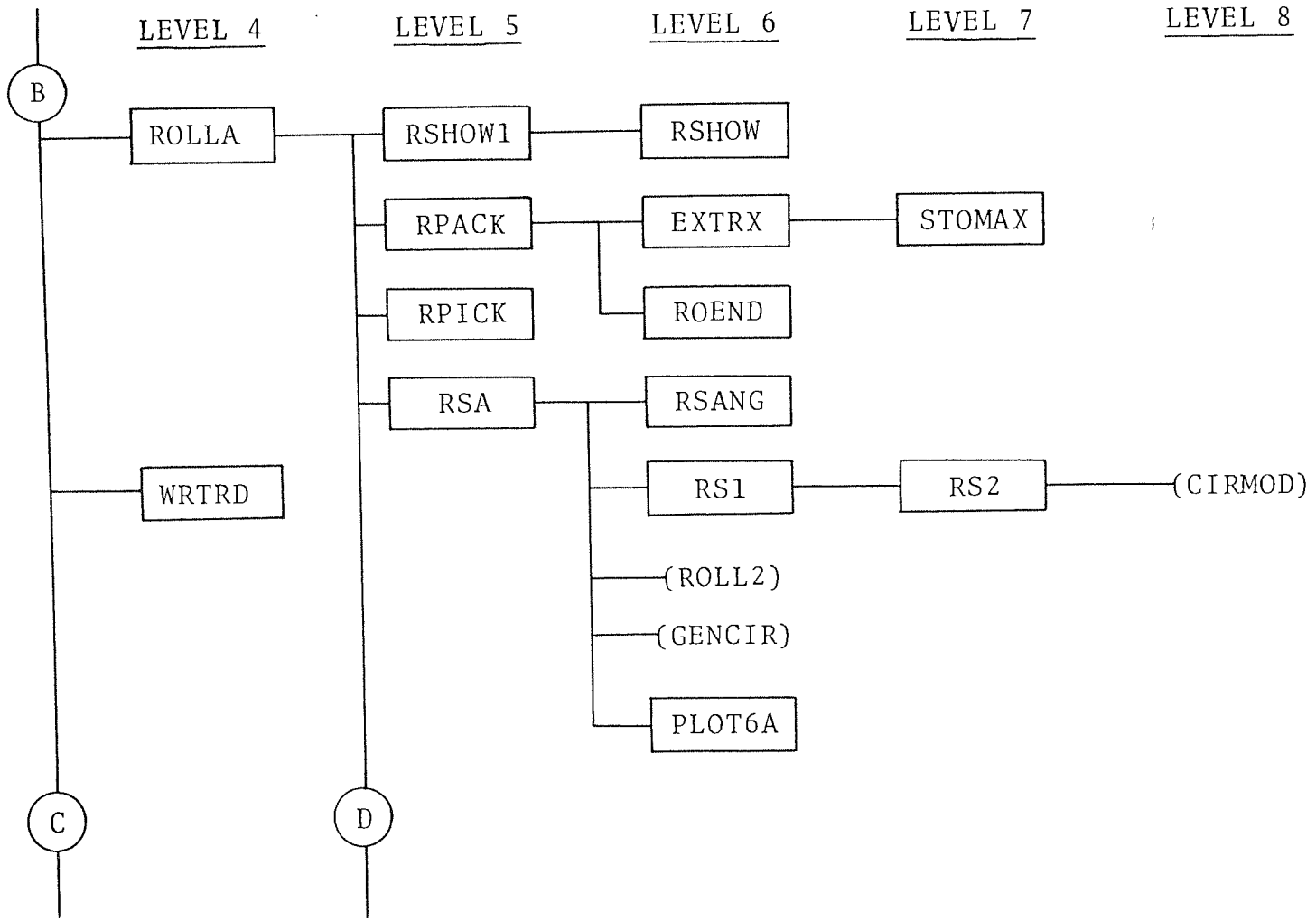


CHART 9.5 (PART 4)

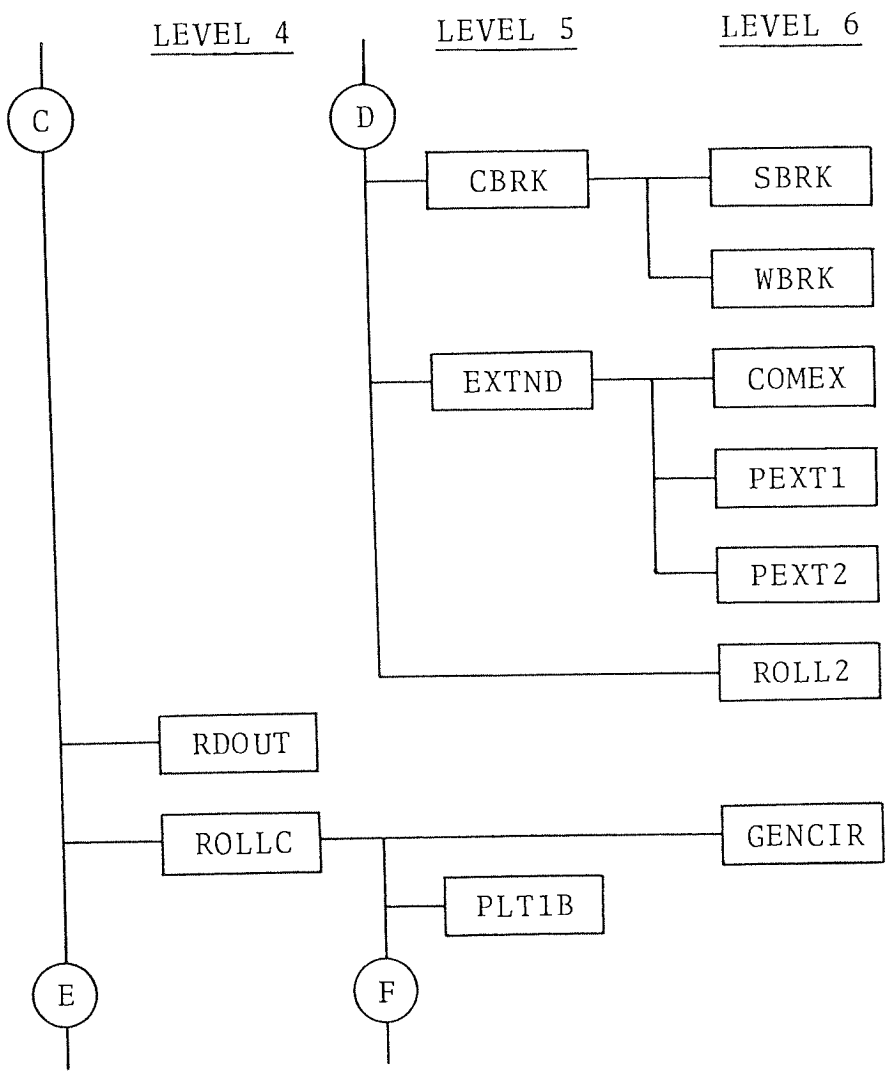


CHART 9.5 (PART 5)

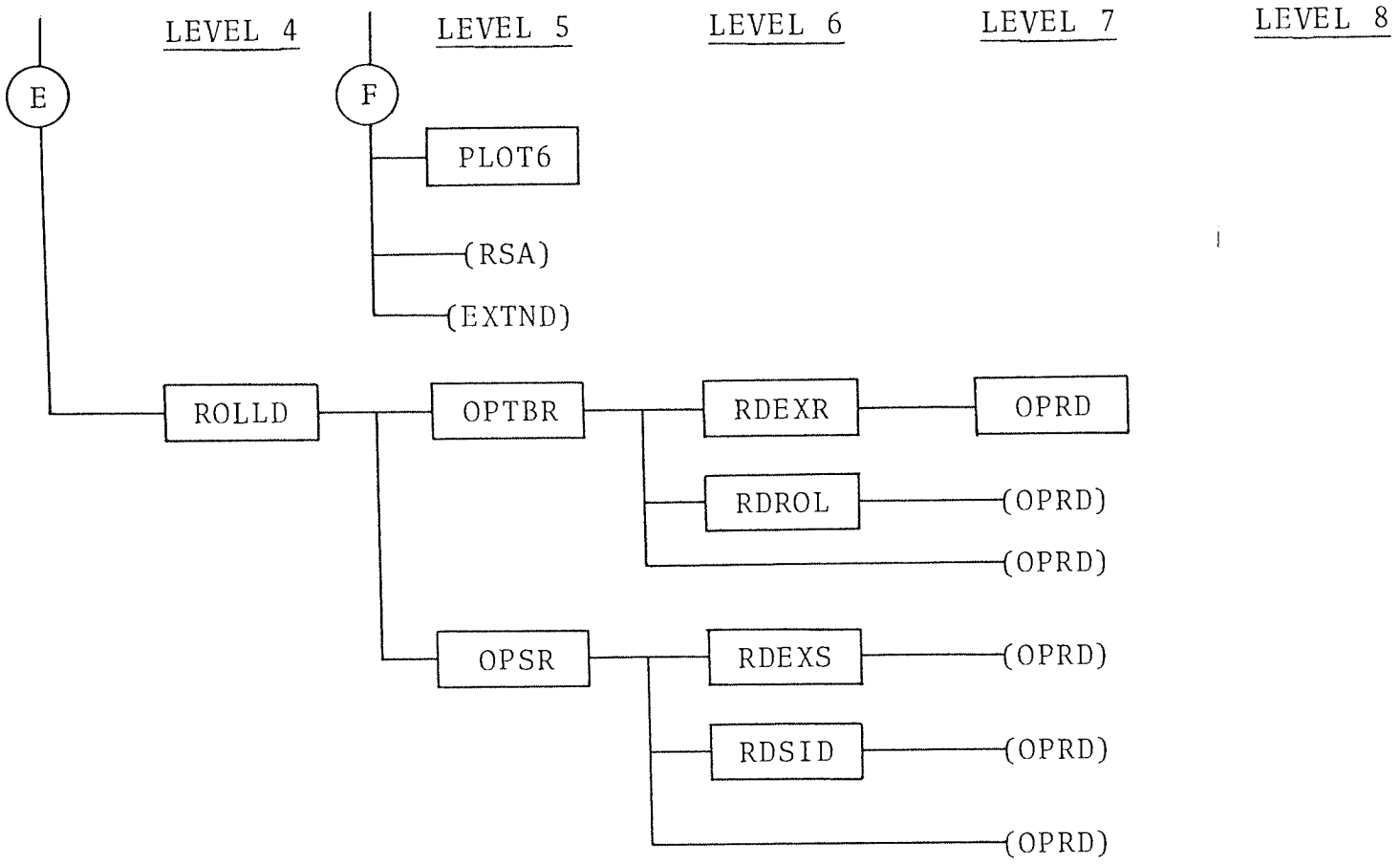


CHART 9.6 HIERARCHY OF THE ROLL INPUT DECODER (DCORL)

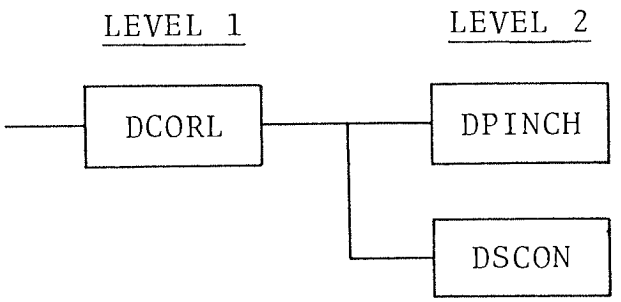


CHART 9.7 LOGIC FLOW OF THE ROLL DESIGN PROGRAM (RPLOT4)

(PART 1)

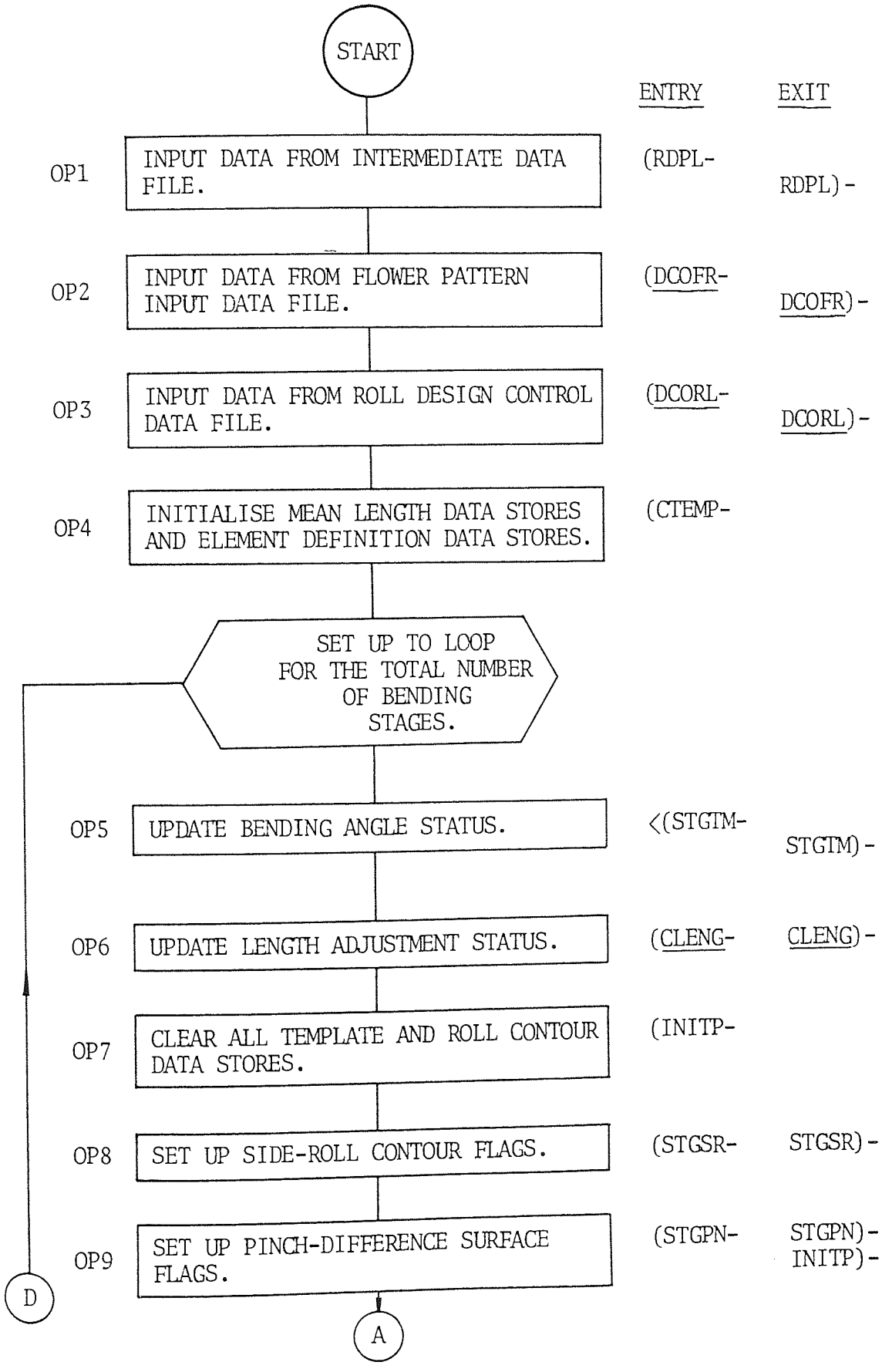


CHART 9.7 (PART 2)

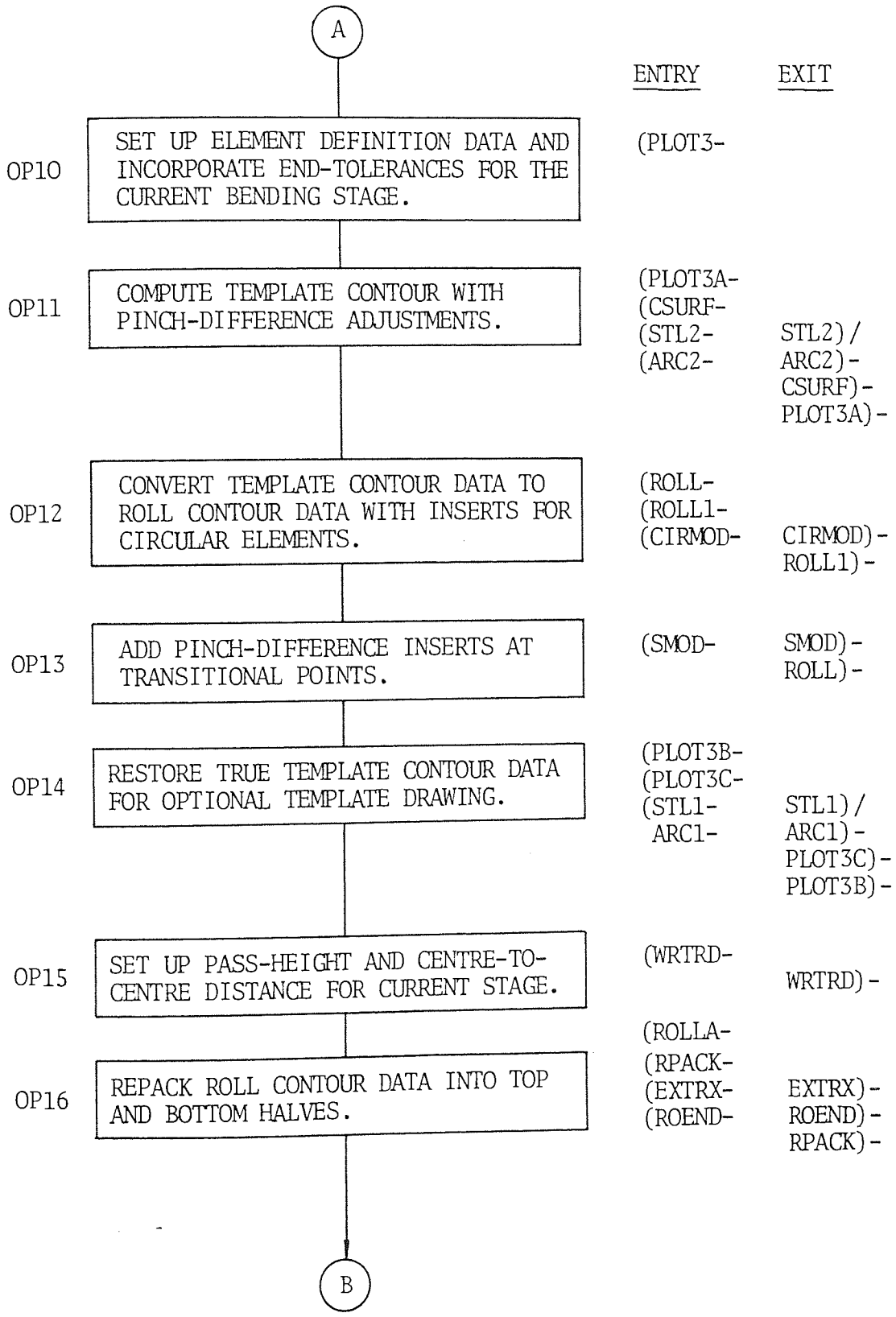


CHART 9.7 (PART 3)

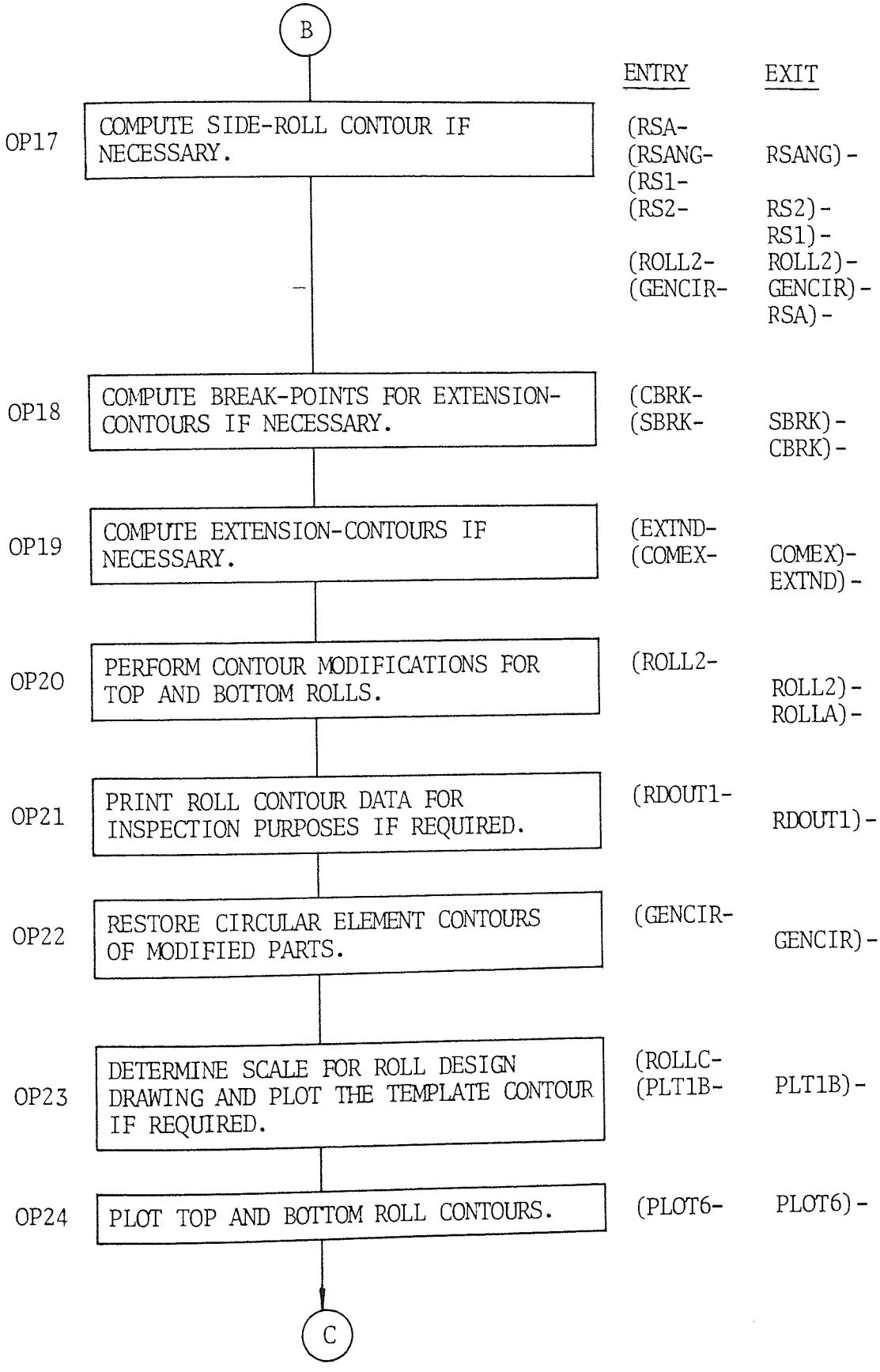
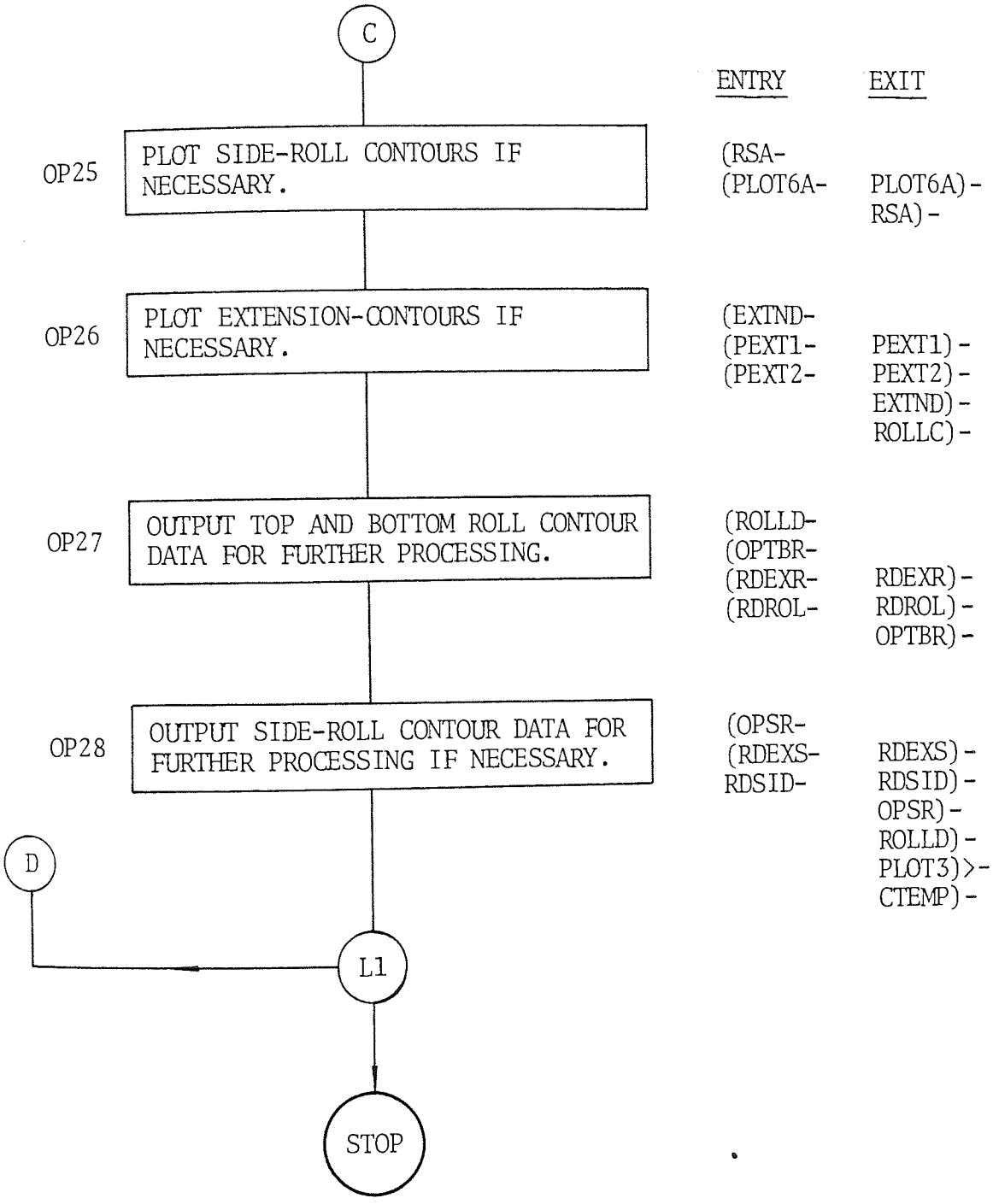


CHART 9.7 (PART 4)



CHAPTER 10

SYSTEM INPUT DATA

10.1 Introduction

As shown in chart 4.1, the input data set for the entire roll design software system has been organised into three distinctive groups. The data required for each group is held in the following designated files:-

(1) DXXXX - The Section Input Data File as the first input data file.

(2) FXXXX - The Flower Pattern Input Data File as the second input data file.

(3) RXXXX - The Roll Input Data File as the third input data file.

(where XXXX is any four alphanumeric characters).

The layout for each of them is as shown in Tables 6.1 to 6.3 respectively and the description of the contents in each of them can be found in Tables 6.4 to 6.5 respectively.

As illustrated in Tables 6.1 to 6.3, all data units in each file are classified into subgroups which are identified by the characteristic letter associated with the file

and the subgroup number, together with an item number in the subgroup separated by a decimal point. For instance, a data unit in the first data file, subgroup 5 and unit 1 will be named as:-

D5.1

Each subgroup may consist of one or more entries of one or more lines, with each line containing one or more data units. The dotted strings represent repetition of the entries in each subgroup and in most cases the repetition is terminated with a line of dummy zero values.

The data units enclosed by hidden lines in the file are optional input data depending on selection. The data types are I for integer, R for real number and A for alphanumeric text.

10.2 The Section Input Data File

The purpose of this data file is to define the finished section for drawing purposes and the generation of data for the intermediate data file (see chart 4.1) for later processing.

The following is a brief description of the manner the data should be entered subgroup by subgroup. Details of the file are in Tables 10.1 and 10.4.

Subgroup D1 deals with the general definition data for the input unit, the output unit, the section or material thickness and the selection of the definition scheme for the section (ORIGIN).

Subgroup D2 deals with defining the section according to the scheme (ORIGIN) selected in terms of elements each defined by an entry of four data units. The entries form either one or two definition sequences depending on whether the section is one-sided or two-sided.

Subgroup D3 deals with the output control for frame size selection and the choice of the desired scale for drawing.

Subgroup D4 deals with the inclusion or exclusion of the dimensioning operation.

Subgroup D5 deals with the inclusion or exclusion of the input for the title-block contents. Subgroup D6 should be skipped if exclusion is the selection.

Subgroup D6 deals with the printing of the desired text strings in the title-block as selected according to Table 6.9.

10.3 The Flower Pattern Input Data File

The purpose of this data file is to define the element

bending and element length monitoring pattern for the generation of the flower patterns, the 10 to 1 templates and the roll designs (see chart 4.1). Details of the file are in Tables 10.2 and 10.5.

Subgroup F1 deals with the selection for the flower pattern output, the roll output or both.

Subgroup F2 deals with the total number of bending stages to be used.

Subgroup F3 deals with whether the Composite Element Length Definition is required or otherwise.

Subgroup F4 deals with defining the bending status for each element in each stage with or without the Composite Length Definition for all stages.

Subgroup F5 deals with the selection of the stages requiring Radii Sharpening operation.

Subgroup F6 deals with the activating and cancelling of the Fixed Percentage Element Length operation for short-leg bending.

10.4 The Roll Input Data File

The purpose of this data file is to define and control the roll designs, with the optional features included if so desired. Details of the file are in Tables 10.3 and

10.6.

Subgroup R1 deals with the data for defining the pass-heights of the bottom rolls and the centre-to-centre distances between the top and bottom rolls.

Subgroup R2 deals with the data for defining the end-tolerances for both the L.H.S. and R.H.S. of the template at each bending stage.

Subgroup R3 deals with the inclusion or exclusion of the template drawing in hidden-line form, the scale control and the relative placing of each roll drawing.

Subgroup R4 deals with the selection of the stages for which roll drawings are required.

Subgroup R5 deals with the selection of the type of pinch-difference definition scheme required.

Subgroup R6 deals with the data for controlling pinch-difference clearance.

Subgroup R7 deals with the data for defining the drive-surface of the rolls at each stage.

Subgroup R8 deals with the inclusion or exclusion of the side-roll option.

Subgroup R9 deals with the data for defining the distances between the centre (ORIGIN) of the template and

side-roll axes.

Subgroup R10 deals with the data for defining the side-roll contours at each stage.

Subgroup R11 deals with the inclusion or exclusion of the extension-contour option.

Subgroup R12 deals with the definition of each extension-contour to be used.

Subgroup R13 deals with the selection of the defined extension-contours at each stage.

10.5 The Processing Commands

With the existing processing arrangements on the ICL 1904s computer system, the following processing commands applies:-

The command for generating the finished section drawing only is:-

SECTION SECXXXX, DXXXX, OUTXXXX

where SECXXXX is the job name,
DXXXX is the name of the first input data file, and
OUTXXXX is the name of the output file.

The command for generating the finished section

drawing and the flower pattern drawing is:-

FLOWER SECXXXX, DXXXX, FXXXX, OUTXXXX

where SECXXXX, DXXXX, OUTXXXX are as for the SECTION command and
FXXXX is the name of the second input data file.

The command for generating the finished section drawing, the flower pattern drawing and the 10 to 1 template drawings is:-

TEMPLATE SECXXXX, DXXXX, FXXXX, OUTXXXX

where SECXXXX, DXXXX, FXXXX and OUTXXXX are as for the FLOWER command.

(The generated template-contour data will be output to a file named SECXXXX-T).

The command for generating the finished section drawing, the flower pattern drawing and the roll drawings is:-

ROLL SECXXXX, DXXXX, FXXXX, RXXXX, OUTXXXX

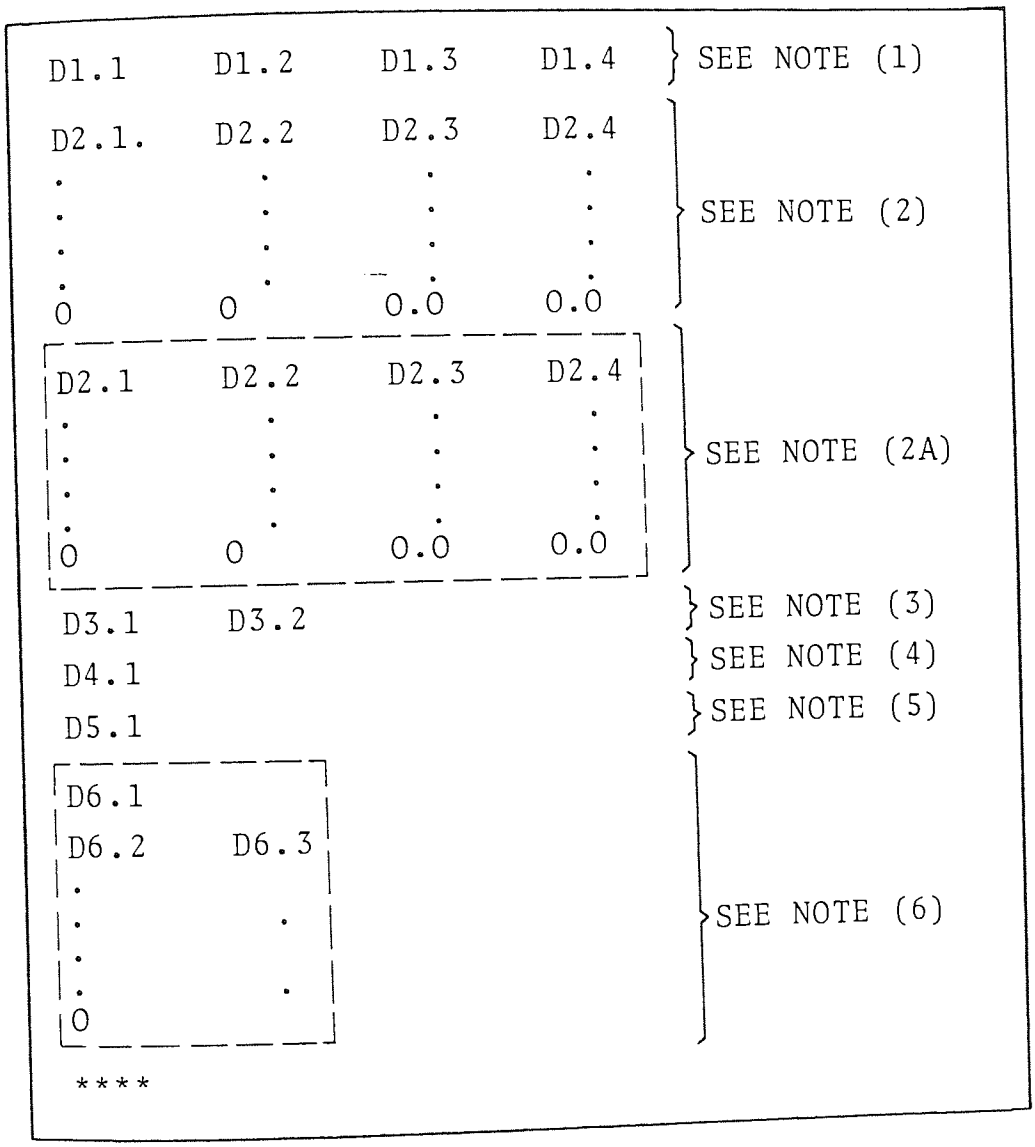
where SECXXXX, DXXXX, FXXXX and OUTXXXX are as for the FLOWER command and
RXXXX is the name of the third input data file.

(The generated roll contour data will be output to a file named SECXXXX-R).

An example of processing using the ROLL command is illustrated in Appendix 1.

TABLE 10.1 LAYOUT OF THE FIRST INPUT DATA FILE

FILE DXXXX



NOTES:-

(1) SUBGROUP D1 - 1 line

(2) SUBGROUP D2 - 1 to 50 lines (1 line per entry)

(2A) SUBGROUP D2 - 1 to 50 lines

(3) SUBGROUP D3 - 1 line

(4) SUBGROUP D4 - 1 line

(5) SUBGROUP D5 - 1 line

(6) SUBGROUP D6 - 2 to 20 lines (2 lines per entry)

TABLE 10.2 LAYOUT OF THE SECOND INPUT DATA FILE

FILE FXXXX

```
F1.1                                                                    } (1)
F2.1                                                                    } (2)
F3.1                                                                    } (3)
F4.1  F4.2  F4.3  F4.4  F4.5  F4.6  F4.7  F4.8  F4.9  F4.10  F4.11      } (4)
  .     .     .     .     .     .     .     .     .     .      .
  .     .     .     .     .     .     .     .     .     .      .
  .     .     .     .     .     .     .     .     .     .      .
  .     .     .     .     .     .     .     .     .     .      .
  .     .     .     .     .     .     .     .     .     .      .
  .     .     .     .     .     .     .     .     .     .      .
  0     0    0.0    0     0     0     0    0.0    0     0      0
F5.1                                                                    } (5)
F5.2
F6.1  F6.2  F6.3  F6.4  F6.5  F6.6  F6.7                               } (6)
  .     .     .     .     .     .     .
  .     .     .     .     .     .     .
  0     0     0     0     0     0     0
****
```

NOTES

(1) SUBGROUP F1 - 1 line

(2) SUBGROUP F2 - 1 line

(3) SUBGROUP F3 - 1 line

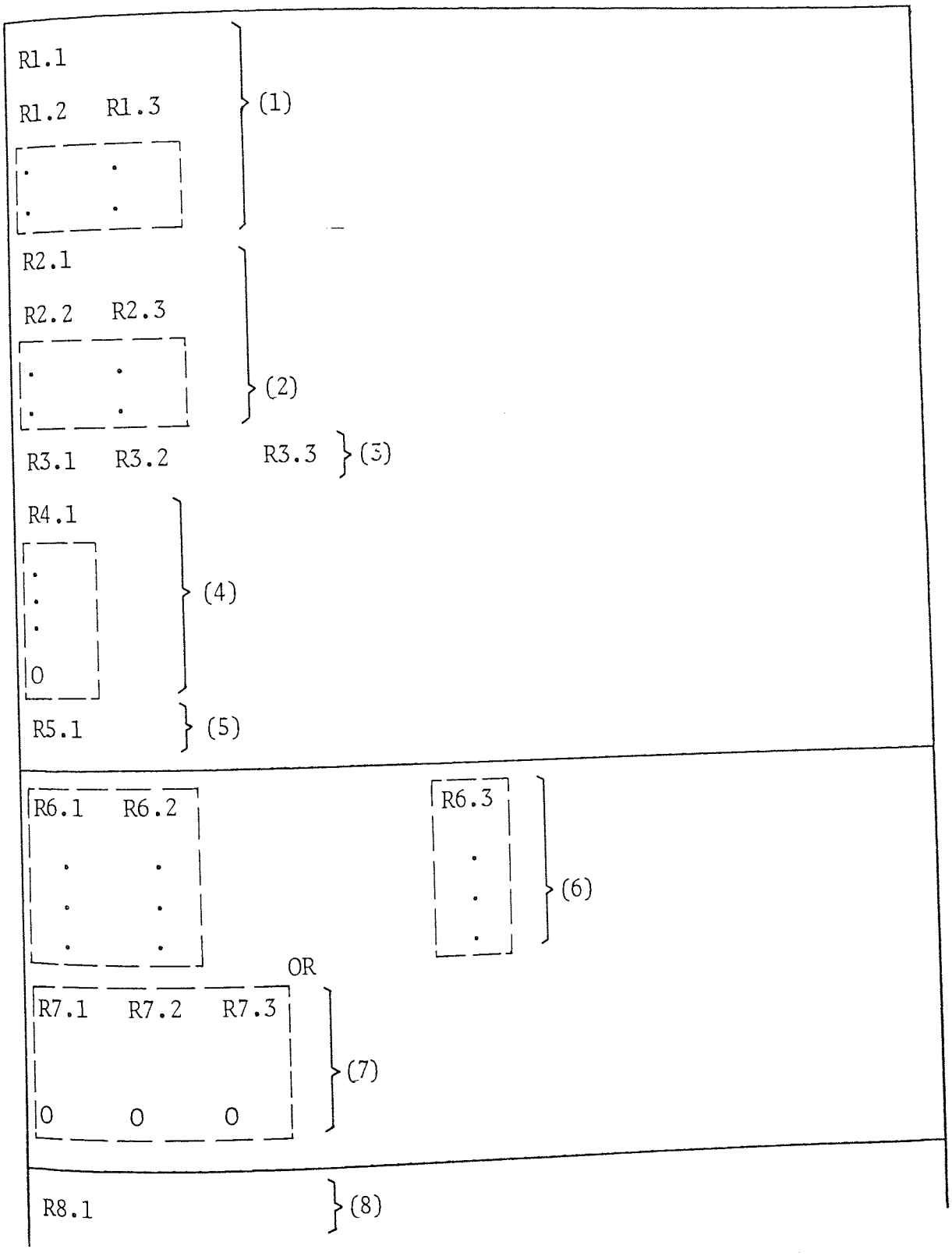
(4) SUBGROUP F4 - 1 to 50 lines (1 line per entry)

(5) SUBGROUP F5 - 1 line plus 0 to 50 lines (1 line per entry)

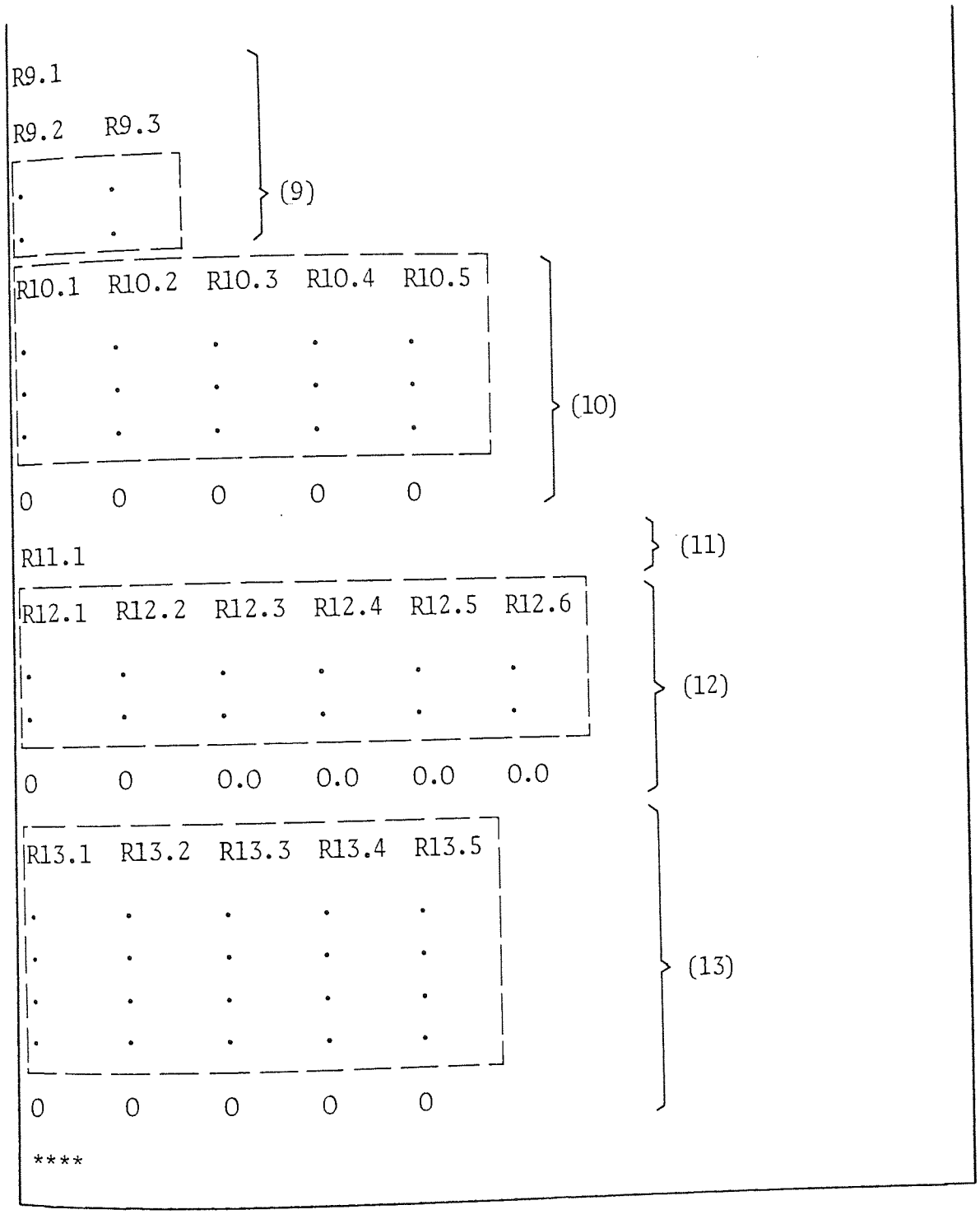
(6) SUBGROUP F6 - 1 to 50 lines (1 line per entry)

TABLE 10.3 LAYOUT OF THE THIRD INPUT DATA FILE

FILE RXXXX



(to be continued)



NOTES

Let N = Total number of stages (see data unit F2.1)

(1) SUBGROUP R1 - 1 line plus 1 or N lines (1 line per entry)

(2) SUBGROUP R2 - 1 line plus 1 or N lines (1 line per entry)

(3) SUBGROUP R3 - 1 line

(4) SUBGROUP R4 - 1 line plus 0 to N lines (1 line per entry)

(5) SUBGROUP R5 - 1 line

(6) SUBGROUP R6 - 1 or N lines (1 line per entry)
(7) SUBGROUP R7 - 1 to 2N lines (2 lines per entry)
(8) SUBGROUP R8 - 1 line
(9) SUBGROUP R9 - 1 line plus 1 or N lines (1 line per entry)
(10) SUBGROUP R10 - 1 to 2N lines (2 lines per entry)
(11) SUBGROUP R11 - 1 line
(12) SUBGROUP R12 - 1 to 50 lines (1 line per entry)
(13) SUBGROUP R13 - 1 to 50 lines (1 or 2 lines per entry)

N.B. The maximum number of lines entered for each subgroup should be 50 or less.

TABLE 10.4 DATA FOR THE FIRST INPUT DATA FILE (PART 1)

THE SECTION INPUT DATA FILE (DXXXX)

ITEM	NAME	TYPE	DESCRIPTION
D1.1	IUNIT	I	Input unit for dimensions (1 for inch and 2 for mm).
D1.2	OUNIT	I	Output unit for dimensions (1 for inch and 2 for mm).
D1.3	THICK	R	Material thickness (positive).
D1.4	ORIGIN	I	Type of element definition scheme (1 if from left to right, 2 if from right to left, 3 if from centre to right, then centre to left, 4 if from centre to left, then centre to right, 5 if from centre to right and symmetrical).
D2.1	IDATA1	I	Element sequence number (start from 1, increment by 1 for each new element, up to 50).
D2.2	IDATA2	I	Element type (1 for linear element and 2 for circular element, 1 if IDATA1 is 1).
D2.3	RDATA1	R	Element length or inside radius, length if linear element and radius if circular element (0 or positive).
D2.4	RDATA2	R	Angle of bending, normally for circular elements only, but if IDATA1 is 1 then it is the angle of inclination for the section (positive for upward bending and negative for downward bending).
D3.1	PSIZE	I	Frame size for the drawing (0 to 4 for A0 to A4 sizes respectively).
D3.2	SCALE	R	Scale for the section drawing (any positive value below the given limit for the selected frame size).
D4.1	IDIMEN	I	Section dimensioning (1 for yes and 0 for no).

TABLE 10.4 DATA FOR THE FIRST INPUT DATA FILE (PART 2)

ITEM	NAME	TYPE	DESCRIPTION
D5.1	ITITLE	I	Input for title-block content (1 if yes and 0 if no, if 0 skip input for subgroup D6).
D6.1	SELNO	I	Number identifying the required part of the title-blocks (1 to 10 as shown in Table 6.9, 0 for termination).
D6.2	NCHAR	I	Number of characters in the input text (not exceeding the shown limit).
D6.3	TEXT	A	Input text to be printed (any alphanumeric characters).

TABLE 10.5 DATA FOR THE SECOND INPUT DATA FILE (PART 1)

THE FLOWER-PATTERN INPUT DATA FILE (FXXXX)

ITEM	NAME	TYPE	DESCRIPTION
F1.1	JRUN	I	Run selection (1 for flower-patterns only, 2 for roll-drawings only and 3 for flower-patterns and roll drawings).
F2.1	NSTAGE	I	Total number of forming stages (1 to 50, subject to the limit by the input data storage space).
F3.1	JBEND	I	Selection of the bending definition scheme (0 for simple element length definition and 1 for composite element length definition, if 0 then items F4.4, F4.5, F4.6, F4.9, F4.10 and F4.11 should be excluded from the bending definitions).
F4.1	ISEQ	I	Current stage number (any positive value equal to or 1 greater than the previous stage number but not exceeding NSTAGE).
F4.2	IELML	I	L.H.S. element number (any element number as defined in subgroup D2, 0 if irrelevant).
F4.3	XANGL	R	L.H.S. angle of bending (positive for upward bending and negative for downward bending).
F4.4	IPCL1	I	Percentage length of leading part of L.H.S. element (0 or positive).
F4.5	IPCL2	I	Percentage length of circular part of L.H.S. element (0 or positive).
F4.6	IPCL3	I	Percentage length of trailing part of L.H.S. element (0 or positive).

TABLE 10.5 DATA FOR THE SECOND INPUT DATA FILE (PART 2)

ITEM	NAME	TYPE	DESCRIPTION
F4.7	IELMR	I	R.H.S. equivalent of IELML.
F4.8	XANGR	R	R.H.S. equivalent of XANGL.
F4.9	IPCR1	I	R.H.S. equivalent of IPCL1.
F4.10	IPCR2	I	R.H.S. equivalent of IPCL2.
F4.11	IPCR3	I	R.H.S. equivalent of IPCL3.
F5.1	JSHARP	I	Selection of Radii-Sharpening definition scheme (0 if Radii-Sharpening is not required, -1 if Radii-Sharpening is required for all stages except the last stage, 1 if Radii-Sharpening is required for stages to be specified subsequently, if 0 or -1 then input for F5.2 is not needed, if 1 then supply as many value as desired for F5.2 and terminate by a zero value).
F5.2	ISQSH	I	Stage number requiring Radii-Sharpening (0 or positive, not exceeding NSTAGE).
F6.1	ISQCP	I	Entry number for Fixed Percentage Length selection (0 or positive, 0 for termination).
F6.2	IECL1	I	L.H.S. element number (any element number as defined in subgroup D2, 0 if irrelevant).
F6.3	IECL2	I	Stage number, starting from which Fixed Percentage Length will be used for IECL1 (positive and less than IECL3).
F6.4	IECL3	I	Stage number when Fixed Percentage Length cease to operate for IECL1 (positive, greater than IECL2 but less than NSTAGE).
F6.5	IECR1	I	R.H.S. equivalent of IECL1.
F6.6	IECR2	I	R.H.S. equivalent of IECL2.
F6.7	IECR3	I	R.H.S. equivalent of IECL3.

TABLE 10.6 DATA FOR THE THIRD INPUT DATA FILE (PART 1)

THE ROLL INPUT DATA FILE (RXXXX)

ITEM	NAME	TYPE	DESCRIPTION
R1.1	IPASHT	I	Selection of definition scheme for pass-height and centre-to-centre distance (1 if one common entry for all stages, -1 if separate entry for each stage).
R1.2	PASHT	R	Pass-height of strip (any positive value).
R1.3	CTOC	R	Centre-to-centre distance between top and bottom rolls (any positive value greater than PASHT).
R2.1	ITOL	I	Selection of definition scheme for end-tolerance (1 if one common entry for all stages, -1 if separate entry for each stage).
R2.2	TOLLH	R	L.H.S. end-tolerance (zero or positive).
R2.3	TOLRH	R	L.H.S. end-tolerance (zero or positive).
R3.1	JTEMP	I	Template drawing in hidden-line form (1 if yes, 0 if no).
R3.2	RSCALE	R	Desired scale for roll drawings (any positive value subject to paper width limit).
R3.3	RGAP	R	Gap between the roll drawings (zero or positive).
R4.1	JSTAGE	I	Selection of stages for roll drawings (-1 if all stages required, 0 if none required and any positive value not exceeding NSTAGE if selected stages required, in the last case supply as many values as desired and terminate with a zero value).

TABLE 10.6 DATA FOR THE THIRD INPUT DATA FILE (PART 2)

ITEM	NAME	TYPE	DESCRIPTION
R5.1	JPINCH	I	Selection of scheme for defining pinch-difference dimensions (0 if no external values are supplied, 1 if only 1 set of values for all stages or -1 if different set of values for each stage, both 1 and -1 are for selecting the "double-thickness" option; 2 if only 1 clearance value for all stages or -2 if separate clearance value for each stage, both 2 and -2 are for selecting the "single-thickness" option; if 0, 1 or -1 is entered, then define values for R6.1 and R6.2 only and proceed to subgroup R7, if 2 or -2 is entered, then define values for R6.3 only and skip subgroup R7).
R6.1	PDIM1	R	Gap between drive-surfaces (any positive value).
R6.2	PDIM2	R	Gap between clearance-surfaces (any positive value greater than PDIM1).
R6.3	CLEAR	R	Clearance to be added to 108 percent of thickness (any positive value).
R7.1	ISQP	I	Stage number (1 to NSTAGE, two entries for each stage).
R7.2	IEPL	I	L.H.S. element number defining the beginning or the end of the drive-surface (any value as defined in subgroup D2, 0 if irrelevant).
R7.3	IEPR	I	R.H.S. equivalent of IEPL.
R8.1	ISROL	I	Selection of side-roll option (1 if yes, 0 if no, if 0 then skip subgroups R9 to R13).

TABLE 10.6 DATA FOR THE THIRD INPUT DATA FILE (PART 3)

ITEM	NAME	TYPE	DESCRIPTION
R9.1	IOTOS	I	Selection of scheme for defining ORIGIN to side-roll centre distances (1 if one common entry for all stages or -1 if separate entry for each stage).
R9.2	OTOSL	R	L.H.S. ORIGIN to side-roll centre distance (any positive value).
R9.3	OTOSR	R	R.H.S. ORIGIN to side-roll centre distance (any positive value).
R10.1	ISQS	I	Stage number (1 to NSTAGE, two entries per stage).
R10.2	IESL1	I	L.H.S. element number defining the beginning or the end of the side-roll contour (any value as defined in subgroup D2, 0 if irrelevant).
R10.3	IESL2	I	Face number for element defined in IESL1 (see section 9.4.2).
R10.4	IESR1	I	R.H.S. equivalent of IESL1.
R10.5	IESR2	I	Face number for element defined in IESR1.
R11.1	IEXTN	I	Selection of the extension-contour option (1 if yes or 0 if no, if 0 skip subgroups R12 and R13).
R12.1	ISC1	I	Extension-contour number (1 to 50).
R12.2	ISC2	I	Extension-contour type (1 for side-roll extension-contour, 2 for spigot type A and 3 for spigot type B).
R12.3	SCDIM1	R	Part 1 dimension of extension-contour (see Table 9.11).
R12.4	SCDIM2	R	Part 2 dimension of extension-contour (see Table 9.11).
R12.5	SCDIM3	R	Part 3 dimension of extension-contour (see Table 9.11).
R12.6	SCDIM4	R	Part 4 dimension of extension-contour (see Table 9.11).

TABLE 10.6 DATA FOR THE THIRD INPUT DATA FILE (PART 4)

ITEM	NAME	TYPE	DESCRIPTION
R13.1	ISQD	I	Stage number (1 to 50, two entries per stage if side-roll used).
R13.2	IELD1	I	Selection of extension-contour type on L.H.S. (as defined in ISC2).
R13.3	IELD2	I	Selection of extension-contour number on L.H.S. (as defined in ISC1).
R13.4	IERD1	I	Equivalent of IELD1 for R.H.S.
R13.5	IERD2	I	Equivalent of IELD2 for R.H.S.

CHAPTER 11

SYSTEM OUTPUT DATA

11.1 Introduction

There are two kinds of output data generated by the roll design software system, namely the drawings and the contour data files. The type of output data desired can be selected through input data and processing commands fully described in Chapter 10. The types of drawings generated by the software have been illustrated in Figs. 4.4 to 4.8 according to the layout shown in Chart 4.1. There are two types of contour data files generated in output:-

(1) The template contour data file (SECXXXX-T).

(2) The roll contour data file (SECXXXX-R).

The contents of both the data files are organised in identical formats to provide consistency in both data reading and handling, either manually or by computers.

Similar to the data units in the input data files, the data units here are also divided into subgroups which are identified with a name carrying a characteristic letter T, a subgroup number and a unit number separated by a decimal point. The data type is indicated either by a letter I (integer) or a letter R (real number). The details of each data unit for both types of files can be found in Table 11.3.

11.2 The Template Contour Data File

This data file is intended as an interim aid in the manual preparation of NC tapes using some form of computer assisted part-programming facilities, such as the APT system or its equivalent. As shown in Table 11.1, the entries enclosed by hidden lines are optional and repetitive parts depending on the number of bending stages involved. A set of two contours, namely the top face contour and the bottom face contours is associated with each stage.

11.3 The Roll Contour Data File

This data file is intended mainly for the automatic preparation of NC tapes in the CAM arrangement. Like the template contour data file, the entries enclosed in hidden lines (Table 11.2) are also optional and repetitive parts depending on the number of bending stages specified for processing. The number of contours associated with each stage range from two to four, depending on whether side-rolls are present or otherwise.

TABLE 11.1 LAYOUT OF THE TEMPLATE CONTOUR DATA FILE

FILE SECXXXX-T

```
T1.1  T1.2  T1.3  T1.4                                                    } (1)
T2.1                                                                      } (2)
T3.1  T3.2  T3.3  T3.4                                                    } (3)
T4.1
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10              } (4)
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
  0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0   0
T4.1
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10              } (5)
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
  0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0   0
T2.1
T3.1  T3.2  T3.3  T3.4
T4.1
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
  .     .     .     .     .     .     .     .     .     .
```

NOTES

(1) General information.

(2) Bending stage detials.

(3) Drawing control details.

(4) Top face contour details.

(5) Bottom face contour details.

*(2) to (5) are repeated for subsequent stages.

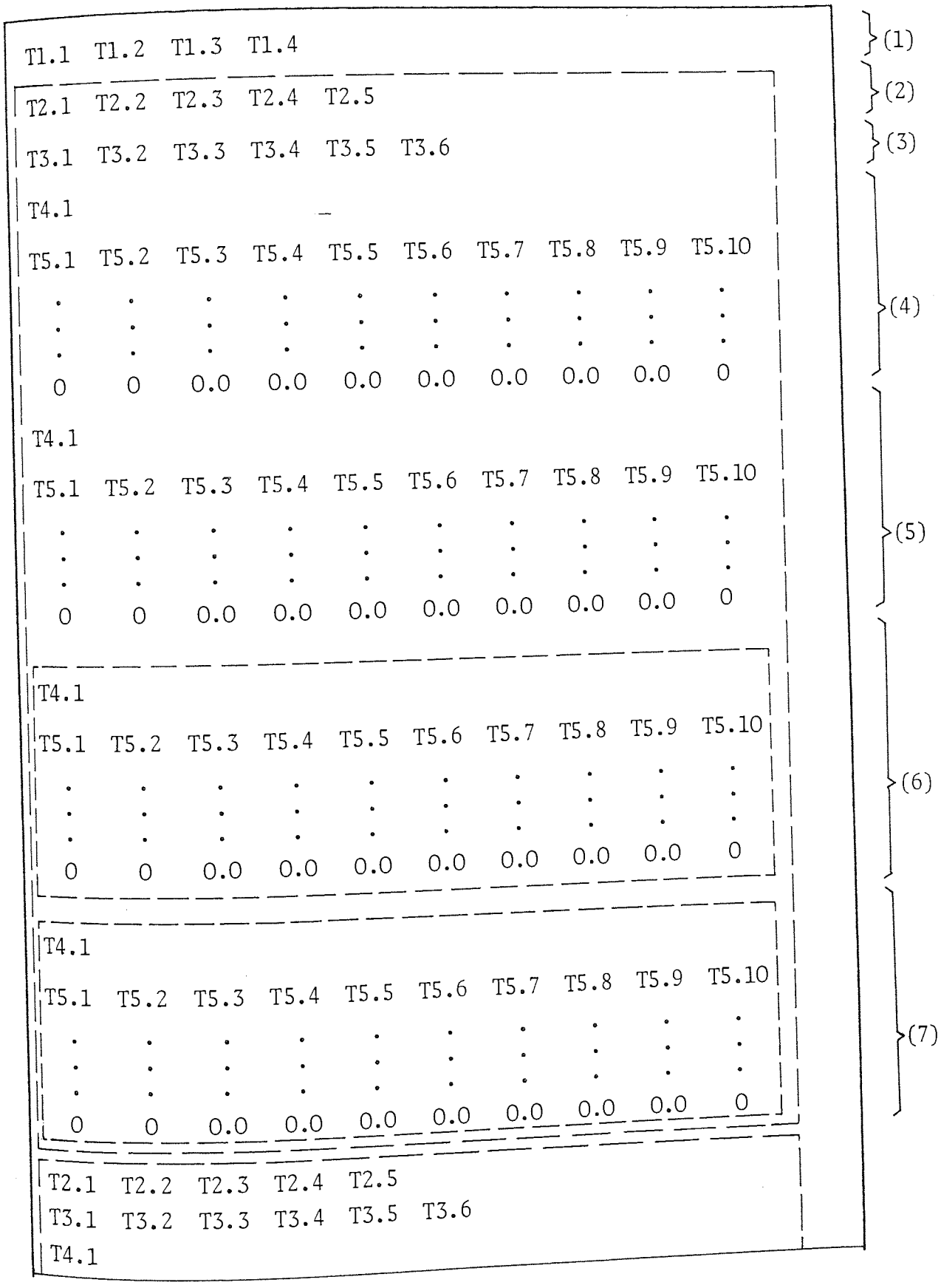


TABLE 11.2 LAYOUT OF THE ROLL CONTOUR DATA FILE

FILE SECXXXX-R

```
T1.1  T1.2  T1.3  T1.4                                                     } (1)
T2.1  T2.2  T2.3  T2.4  T2.5                                               } (2)
T3.1  T3.2  T3.3  T3.4  T3.5  T3.6                                         } (3)
T4.1                                                                       }
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10                }
 .     .     .     .     .     .     .     .     .     .                   } (4)
 .     .     .     .     .     .     .     .     .     .                   }
 .     .     .     .     .     .     .     .     .     .                   }
 0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0    0                   }
T4.1                                                                       }
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10                }
 .     .     .     .     .     .     .     .     .     .                   } (5)
 .     .     .     .     .     .     .     .     .     .                   }
 .     .     .     .     .     .     .     .     .     .                   }
 0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0    0                   }
T4.1                                                                       }
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10                }
 .     .     .     .     .     .     .     .     .     .                   } (6)
 .     .     .     .     .     .     .     .     .     .                   }
 .     .     .     .     .     .     .     .     .     .                   }
 0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0    0                   }
T4.1                                                                       }
T5.1  T5.2  T5.3  T5.4  T5.5  T5.6  T5.7  T5.8  T5.9  T5.10                }
 .     .     .     .     .     .     .     .     .     .                   } (7)
 .     .     .     .     .     .     .     .     .     .                   }
 .     .     .     .     .     .     .     .     .     .                   }
 0     0    0.0   0.0   0.0   0.0   0.0   0.0   0.0    0                   }
T2.1  T2.2  T2.3  T2.4  T2.5
T3.1  T3.2  T3.3  T3.4  T3.5  T3.6
T4.1
```

NOTES

(1) General information.

(2) Bending stage details.

(3) Drawing control details.

(4) Top roll contour details.

(5) Bottom roll contour details.

(6) Optional left side-roll details.

(7) Optional right side-roll details.

*(2) to (7) are repeated for subsequent stages.

TABLE 11.3 DATA OF THE CONTOUR DATA FILES (PART 1)

FILES SECXXXX-T AND SECXXXX-R

ITEM	NAME	TYPE	DESCRIPTION
T1.1	PIE	R	(3.14159265).
T1.2	IUNIT	I	Input unit for dimensions (1 for inch and 2 for mm).
T1.3	OUNIT	I	Output unit for dimensions (1 for inch and 2 for mm).
T1.4	ITOTAL	I	Total number of stages for which sets of roll drawings have been generated.
T2.1	ISTAGE	I	Stage number associated with each set of roll drawings.
T2.2	PASHT	R	The pass-height of the bottom roll at the stage.
T2.3	CTOC	R	The centre-to-centre distance between the top and bottom rolls at the stage.
T2.4	OTOSL	R	The distance between centre of template and left side-roll axis.
T2.5	OTOSR	R	The distance between centre of template and right side-roll axis.
T3.1	XDATUM	R	The X-datum as measured from the extreme left of the paper.
T3.2	ORGXP	R	The distance of translation in X for origin shifting.
T3.3	ORGYP	R	The distance of translation in Y for origin shifting.
T3.4	SCALE	R	Scale for the drawings.
T3.5	ILEFT	I	Presence of left side-roll (1 for yes and 0 for no).
T3.6	IRIGHT	I	Presence of right side-roll (1 for yes and 0 for no).

TABLE 11.3 DATA OF THE CONTOUR DATA FILES (PART 2)

ITEM	NAME	TYPE	DESCRIPTION
T4.1	ICON	I	Roll contour type (1 for top roll, 2 for bottom roll, 3 for left side-roll and 4 for right side-roll) or template contour type (1 for top face and 2 for bottom face contour).
T5.1	NCON	I	Sequence number for element contour (positive, zero for termination of the contour).
T5.2	ITYPE	I	Type of element contour (1 for linear and 2 for circular).
T5.3	XS	R	X-coordinate of starting-point.
T5.4	YS	R	Y-coordinate of starting-point.
T5.5	XE	R	X-coordinate of end-point.
T5.6	YE	R	Y-coordinate of end-point.
T5.7	XC	R	X-coordinate of centre-point (circular element contours only).
T5.8	YC	R	Y-coordinate of centre-point (circular element contours only).
T5.9	R	R	Radius (for circular element contours).
T5.10	IDIR	I	Direction of arc (1 for anticlockwise and -1 for clockwise). Note that all coordinates are absolute and in mm .

CHAPTER 12

CONCLUSIONS AND FUTURE WORK

12.1 Conclusions

The work undertaken in this research has successfully created the required computer aided tools by software development for the design of form-rolls and for the automatic generation of roll contour data needed for roll manufacturing using NC lathes. Based upon all aspects of the work, the following results and conclusions are obtained.

1. According to the needs of the company, the following four major areas of interests were identified as promising for computerisation purposes, forming the scope of investigation into the conventional roll design and manufacture:

   (i) The drafting of the finished section.

   (ii) The design and drafting of the flower patterns.

   (iii) The design and drafting of the 10 to 1 templates.

(iv) The design and drafting of the rolls.

Systematic investigation revealed that the most inefficient tasks in these areas are those which were tedious, error-prone, time consuming and uninteresting when performed entirely by manual effort. Typical examples were routine design decisions, calculations and drafting. Long lead time and low work efficiency were common and were mainly caused by the slowness of the manual process, by the necessary and frequent repetition of design procedures as a result of design mistakes or negligence and by the inconsistent human performance being affected by both the physical and mental state during work.

The existing approach in designing was mainly based upon personal experience, resulting in the presence of many inconsistent and individual styles of designing. Standardised and precise procedures of designing were absent, only approximate guides were used.

2. In assessing the suitability of the existing methods for computerisation, it was found that each routine constituent of the methods could be divided into three distinctive parts for precise analysis:

(i) Set of conditions on which the design decisions and actions are based.

(ii) Set of rules matching each design action to each given situation or each given combination of conditions.

(iii) Set of alternative or mutually exclusive design actions to be performed.

Unless these parts are definable, they cannot be automatically handled by computers. Those routine constituents which are partially definable can thus only be partially computerised. The indefinability was mainly due to the nature of the conventional design practices, practices which were often based on experience accumulated through long process of trial and error and which could often be subjective and personal, lacking in scientific

approach. In situations where there are more than one set of conditions and one set of decision rules for the same design task, then standardisation appears to be the best solution. Standardisation can normally be accomplished through consensus among the designers or in some cases, through the development of new but equivalent methods based on scientific principles. In circumstances where standardisation proved impossible, then the alternative solution which could still be attempted would be to include in the software all equally valid design actions for selection by the designers themselves, to accomplish a limited degree of computerisation. Consistency and definability in design conditions, decisions and actions are therefore the vital prerequisites for computerisation. Suitability for computerisation of a process is also subject to the possibility of establishing an equivalent process that computers can manage, in terms of technical feasibility.

3. Among the various practicable approaches of computerisation which had been examined against the currently available technology, it was found that the most suitable approach was by designing and developing special application software to be implemented on standardised computer systems, based upon the company's requirements and existing interests. Such an approach permits high flexibility in the selection of equipment for implementing the computerised methods. It also permits continuous and close liason with the company throughout the development of software which will precisely meet the company's requirements, existing interests and development plans. The following research programme was planned accordingly for the company, in three distinctive stages of software development:

First Stage

Design and development of software for automatic drafting of the finished section, the flower patterns and the 10 to 1 templates, incorporating the required design definitions and control features. Roll manufacture by

conventional methods using wire-templates continues.

Second Stage

Design and development of the roll design software incorporating as many routine design features as feasible to automatically generate roll drawings and roll contour data to be further processed for NC manufacture. Roll manufacture by NC begins, initially using manual NC tape-preparation process with computer-assisted part-programming as an interim measure. Work load is then to be gradually transferred from conventional manufacture to NC manufacture.

Third Stage

Design and development of software facilities for random roll-contour editing and automatic preparation of NC tapes for machining the rolls. Probably this CAM phase will then be linked to the CAD phase to form an integrated system.

Both the first stage and the second stage development covering the major aspects of roll designing have been completed in this research. The company at the time of writing is implementing the second stage development plan, involving switching from the use of conventional lathes to NC lathes for roll manufacture.

4. It was found that computerisation requirements or needs in general could be precisely defined in terms of the software system characteristics because the nature of the software process corresponds to an input-output system. The developed software has been designed primarily for reliability which is vital to its performance in service, portability which facilitates its easy transfer to a different operating environment and flexibility which is essential for modification and maintenance purposes. Reliability has been maximised through proper program and data structuring, good coding style, systematic development and comprehensive testing. Portability has been accomplished by writing the software in FORTRAN, which is a widely supported high level

language in most computer systems, and by adhering closely to the ANSI standard. Flexibility has been achieved mainly through modular structuring with "single-entry and single-exit" subprograms broadly based on Structured Programming principles. Although the GINO-F software library has been used for drawing purposes, switching over to other graphic facilities is not troublesome, requiring only minor modifications to the developed software.

5. Software development in this research has been successfully and efficiently carried out using the Data Flow Analysis technique, a technique which evolved from this work. The technique is based upon streaming the data flow of the software system through data structuring, with the function to be executed in each stream decomposed into subfunctions in series. Each subfunction is then designed, tested and progressively integrated in the natural sequence of execution, using the input data set to the stream as the sole test data source and using the intermediate data sets for error detection

and removal purposes. The technique permits simultaneous development of software functions in independent data streams branch by branch instead of level by level of the software hierarchy. It also permits effective and systematic testing, debugging and integration resulting in a high degree of software reliability. The technique has been found to be complementary to Structured Programming.

6. While designing the definition scheme for the finished section, it was found that only linear elements and circular elements with uniform thickness were sufficient for the purpose, conforming to the actual design practices. It was also found that the geometry of the section may be precisely defined with a cumulative angle scheme and that it may be computed using two general equations as follows:

$$X_B = X_A + 1 \cos \theta$$

$$Y_B = Y_A + 1 \sin \theta$$

Where $P_A(X_A, Y_A)$ is the reference point of the

element contour of the required geometry, while $P_B(X_B, Y_B)$ is the derived point, θ is the cumulative angle of inclination or orientation and 1 is the distance between the two points.

In calculationg the meanlength for circular elements, it was found that the most generally used method was given by the following equation:

$$r_m = r + kt$$

Where r_m is the mean radius measured from the centre to the neutral-axis of the element, r is the inside radius, t is the element thickness and k is an arbitrary factor.

The most controversial point has been the determination of the value of k, which is approximate in the generally used formulae, so because of the inconsistent behaviour of the elements in thinning during forming.

By geometrical analysis, assuming constant material volume throughout forming, which is generally valid, it was found that the value of

k for t to be uniform or without thinning before and after the element bending should be exactly 0.5, that in turn means the corresponding mean radius will always be:

$$r_m = r + 0.5t$$

If t could be maintained constant throughout bending, the equation will always be valid for precise calculation of the meanlength for circular elements. That has led to the introduction and use of the Opening-Radii method of forming, plus its variations, for preventing or at least reducing the tendency of thinning of circular elements during forming.

7. While designing the procedure for constructing the flower patterns for forming sequence design, it was found that the control of the bending radii in relation to the bending angles could be based upon the principle of constant cross-sectional area of the bent elements, to maintain uniform thickness throughout forming. The Opening-Radii method of forming was the first of several

methods to be so designed and used successfully. It was based upon keeping a constant arc length of each bent element by varying the bending radii inversely proportional to the bending angles at successive forming stages. The only limitation of the method is that when the use of very sharp bending radii is required for overcoming material springback and slipping, such as in the case of bending very short legs, the larger initial radii generated by the method are unsatistactory. The problem was solved by introducing the Composite Element Length Definition method, which was still based upon the principle of constant cross-sectional area, but the circular element is to be defined as a composite element made up of a circular part enclosed by two linear parts at both ends, thereby achieving sharper radii of bending without sacrificing the uniform thickness during forming. A special Radii-Sharpening method has also been introduced to slightly sharpen each bending radius to improve the wear characteristics as well as the control of pinch-difference clearance of the rolls designed. All the described methods are also

applicable to the design of templates and rolls.

8. The 10 to 1 template software has been developed to automatically generate the 10 to 1 template drawings and template contour data for more efficient production of wire-templates while roll manufacturing-using conventional machines is still indispensible. Desirable features needed in adjusting template element length to compensate for the effect of rolling action on material during forming have also been successfully incorporated.

9. The development of the roll design software constituted a substantial part of this research, mainly because of the size and complexity of the software necessary to accommodate the various facilities needed for roll design in practice. Four major aspects of roll design have been successfully incorporated in the developed software, they are:

(1) BASIC ROLL CONTOUR DESIGN

(ii) THE PINCH-DIFFERENCE OPTION

(iii) THE SIDE-ROLL OPTION

(iv) THE EXTENSION-CONTOUR OPTION

In designing the algorithm for generating the basic roll contours, it was found that template contours could be used as the basis for contructing roll contours as they are both closely related. Owing to the inability of computers in deriving roll contours from template contours in the normal human way of vision, pattern-recongition and thought, a special computerised method equivalent to the manual method had to be created. The computerised method has been successfully designed by using a Basic Roll Contour Model which is suitable for contour conversion and modification processes, thereby accomplishing automatic generation of basic roll designs. The required conversion and modification are carried out according to recognizable conditions which represent the various contour patterns. The main difficulty was that because of the wide variety of template shapes involved, the procedures necessary to identify the types of intricate patterns warranting contour modifications became very

complicated to be made fool-proof under all likely circumstances. Nevertheless, an adequate number of procedures has been designed to handle most of the shapes normally encountered in practice, under the constraints imposed by the existing limits on software size and complexity.

In designing the algorithm for generating pinch-difference surfaces, it was found that a pinch-difference surface model with only two kinds of roll surfaces, namely the drive-surface and the clearance-surface, was sufficient to provide the required clearance control at the elements of the strip being formed.

In designing the algorithm for generating side-roll contours, it was found that side-roll contours require the same kind of processing as the basic top and bottom roll contours, but at a different orientation. It was also found that by making the side-roll contour model identical to the basic roll contour model, duplication of the existing procedures could be and was avoided, thereby saving an appreciable amount of software space.

In designing the definition scheme for the extension-contours, it was found that the types of extension-contours used in practice could be adequately covered and precisely defined in the form of a contour consisting of four linear parts of fixed orientations.

With the described capabilities of the roll design software, automatic generation of the complete form of roll designs and the corresponding roll contour data is very much faster and more consistent than the conventional methods, permitting the use of NC machines in roll manufacturing.

10. The developed software system has been successfully implemented on the ICL 1904 S computer at the University and has also been made available for usage by the company personnels in their design work via a remote VDU terminal linked to the University by a dialed-up GPO line, on a temporary basis. The results have been satisfactory in terms of software performance and beneficial in terms of improvement in lead time and productivity in designing. During the interim period, the

10 to 1 template software has been frequently relied on for producing drawings required to manufacture wire-templates while conventional roll manufacturing is still in use, thereby fulfilling the short-term interests, probably until the switch over to the use of NC machines is completed. Continuous liason with the company has brought about constant improvement of the developed software, more extensive standardisation of the existing design practices and the accomplishment of more consistent roll designs, progressively enhancing the general confidence and morale of the design staff in the company. The developed software has also been regularly extended to include additional design features in perfecting the roll designs for better performance in forming otherwise not feasible with conventional means. When the roll manufacturing procedures using NC lathes are fully established, the roll design software will become the key link in generating the essential roll design data for NC tape-preparation purposes. The long-term interests of the company in terms of progress in roll designing techniques and significantly improved productivity are therefore assured. With the various beneficial

results already accomplished, it is thus concluded that computerisation by software development in the design of form-rolls for NC manufacture is viable, in terms of both short-term and long-term benefits. The work also demonstrated convincingly the promising and flexible nature of the CAD/CAM approach towards modernisation and automation.

12.2 Future Work

Most of the essential aspects of the form-roll design methods have been successfully computerised in this research. There is however still room for further improvement in attaining a greater degree of automation for higher work efficiency. For instance, further investigation may be undertaken to establish a more systematic way of designing the forming sequence rather than the traditional trial and error approach. It should be possible to study the relation between the distribution of bending angles at successive forming stages and smooth forming. Similarly, automatic adjustment of pass-heights for the strip according to the severity of bending from station to station may also be established. Another practical area which may be investigated is the automatic

compensation of material springback and slipping during forming.

To meet the continuing needs of the company, software facilities for random and manually performed roll contour editing should either be acquired or developed. Interactive graphic display of the computer generated roll designs using an integrated graphic system is probably the best approach, permitting quick but minor refinements on the roll designs. As a short-term measure, acquiring an NC tape-preparation system involving short-hand part-programming facilities is the best way of linking the already completed CAD stage and the NC manufacturing stag. Meanwhile, further software can be developed to automatically convert the roll design data to NC tape data, forming the CAM stage. Like computerisation for the roll design methods, extensive standardisation in roll manufacturing procedures for all possible kinds of roll shapes has first to be accomplished. Vital aspects are the selecetion of suitable and standardised tooling for efficient machining and work holding, compilation of a technological data base for the correct selection of cutting speeds, feedrates and depth of cuts for various roll materials and machining operations as well as the proper roughing and

finishing sequence for machining the rolls.

Having completed the CAM stage, the possibility of linking it on-line to the CAD stage to form an integrated system is assured. Software for automatic work scheduling and monitoring, parts allocation and machine loading under controlled priority, tool and material stock control and other facilities may then be acquired or developed for production planning and control purposes, if necessary. The final system may then consist of a supervisory computer housing the planning and control software, linking the CAD system and the CAM system together, supplying up-to-date information in all aspects of the work, either in progress or being scheduled, and facilitating more efficient decision making in works management. The system should adequately meet the demand of the competitive future. The CAD/CAM approach will still provide the desired flexibility of selecting areas for stage by stage development according to the balance between needs and capability, without upsetting work continuity.

APPENDIX 1

AN ILLUSTRATION OF TYPICAL OUTPUT LISTINGS GENERATED BY THE COMPUTER DURING THE JOB RUN

1. FILE OUT9999

Listing of all input data and the processing outcome.

2. FILE SEC9999-T

Listing of the template contour data generated with the TEMPLATE command.

3. FILE SEC9999-R

Listing of the roll contour data generated with the ROLL command.

(NB: Figs 4.4 to 4.8 are the drawings generated during the same job run).

```
#LISTING OF :EPP3136.OUT9999(1/)     PRODUCED ON 27OCT81 AT 11.11.10

#68.64AB AT ASTON     IN :EPP3136.MOP1° ON 12NOV81 AT 10.24.06 USING  U15

DOCUMENT     OUT9999


<24@STARTED :EPP3136,MOPSEC9999,27OCT81 10.59.28 TYPE: BACK  IO ; STREAM D ; PRIORITY 2 ; ENTERED WELL 27OCT81 10.19.21
008010.59.28* RJ MOPSEC9999,:EPP3136.ROLLDRAWZ,PARAM(SEC9999,D9999,F9999,R9999,OUT9999),JD(JT600,MZ86K)
000@10.59.28 JOB IS NOW FULLY STARTED
008010.59.28* ROLLDRAWZ        SEC9999,D9999,F9999,R9999,OUT9999
008010.59.29* LF D9999,NU
0004       0 2     2     2.0     3
0004       1 1     1     15.0     0.0
0004       2 2     2     2.0     90.0
0004       3 3     1     10.0     0.0
0004       4 0     0     0.0     0.0
0004       5 1     1     15.0     0.0
0004       6 2     2     2.0     -90.0
0004       7 3     1     15.0     0.0
0004       8 0     0     0.0     0.0
0004       9 4     1.0
0004      10 1
0004      11 1
0004      12 1
0004      13 20 ILLUSTRATION EXAMPLE
0004      14 6
0004      15 6 K.H.NG
0004      16 7
0004      17 7 18.6.81
0004      18 0
0004      19 ****
0004      20
008010.59.30* LF F9999,NU
0004       0 3
0004       1 5
0004       2 0
0004       3 1     2     -35.0     2     35.0
0004       4 2     2     -60.0     2     60.0
0004       5 3     2     -75.0     2     75.0
0004       6 4     2     -85.0     2     85.0
0004       7 5     2     -90.0     2     90.0
0004       8 0     0     0.0     0     0.0
0004       9 -1
0004      10 0     0     0     0     0     0     0
0004      11 ****
0004      12
```

```
0080 10.59.32+ LF R9999,NU
0004        0 1
0004        1 100.0     200.0
0004        2 1
0004        3 1.0      1.0
0004        4 1      1.0      10.0
0004        5 -1
0004        6 0
0004        7 1      1      1
0004        8 1      1      1
0004        9 2      1      1
0004       10 2      1      1
0004       11 3      1      1
0004       12 3      1      1
0004       13 4      1      1
0004       14 4      1      1
0004       15 5      1      1
0004       16 5      1      1
0004       17 0      0      0
0004       18 1
0004       19 1
0004       20 70.0      70.0
0004       21 2      3      4      2      1
0004       22 2      2      2      3      4
0004       23 3      3      4      2      1
0004       24 3      2      2      3      4
0004       25 4      3      4      2      1
0004       26 4      2      2      3      4
0004       27 5      3      2      2      1
0004       28 5      2      2      3      1
0004       29 0      0      0      0      0
0004       30 1
0004       31 1      1      10.0      10.0      20.0      10.0
0004       32 2      2      10.0      10.0      10.0      10.0
0004       33 3      3      15.0      15.0      15.0      15.0
0004       34 0      0      0.0       0.0       0.0       0.0
0004       35 1      3      3      2      2
0004       36 2      1      1      1      1
0004       37 2      1      1      1      1
0004       38 3      1      1      1      1
0004       39 3      1      1      1      1
0004       40 4      1      1      1      1
0004       41 4      1      1      1      1
0004       42 5      1      1      1      1
0004       43 5      1      1      1      1
0004       44 0      0      0      0      0
0004       45 ****
0004       46
```

```
008010.59.35+ UAFORTRAN LOAD :FPP3136.RPLOT1Z-SV,DATA D9999,*LP1 PLOTDATA,LINES 10000,*LPO F1X,EXIT
0080ENTERING :MACROS.UAFORTRAN(227/)
008010.59.36+         TA AB,CM,JL
424011.00.07 0.04 CORE GIVEN 32768
0008DISPLAY : TRACE 2 USED
00080.04 :HALTED : 94
0001DISPLAY: UAFORTRAN: GRAPHICAL OUTPUT FILE IS UACALW2380
0008DISPLAY : GINO-F MK 2.6
0008ALLOT        *CRO     N
00080.06: MONITOR
404011.01.23 FREE *LP1 ,233 TRANSFERS
404011.01.23 FREE *CRO ,19 TRANSFERS
404011.01.23 FREE *LPO ,203 TRANSFERS
00080.29 :DELETED : 00
404011.01.23 FREE *FH7 ,208 TRANSFERS
404011.01.23 0.29 DELETED,CLOCKED 0.23
0001DISPLAY: UAFORTRAN: NORMAL EXIT
008011.01.25+ LF F1X
0004
0004
0004
0004                                   * PLOTTING PROGRAM NO. 1 FOR ROLLED-SECTION *
0004
0004                                                FINISHED SECTION
0004                                                ----------------
0004
0004
0004
0004
0004
0004                    PLEASE SUPPLY THE FOLLOWING DATA ACCORDING TO THE GIVEN FORMAT:-
0004
0004
0004                    INPUT UNIT          OUTPUT UNIT          THICKNESS          ORIGIN
0004                    ----------          -----------          ---------          ------
0004
0004                    (INTEGER)           (INTEGER)            (REAL)             (INTEGER)
0004
0004
0004
0004                    WHERE,
0004
0004                    INPUT UNIT IS IN INCH(1) OR MM(2),
0004                    OUTPUT UNIT IS ALSO IN INCH(1) OR MM(2),
0004                    THICKNESS IS THICKNESS OF STRIP,
0004                    ORIGIN IS THE STARTING POINT FOR ELEMENT DEFINITION SEQUENCE LATER ON.
```

```
ORIGIN=1 IF FROM LEFT TOWARDS RIGHT (ONE-SIDED).
ORIGIN=2 IF FROM RIGHT TOWARDS LEFT (ONE-SIDED).
ORIGIN=3 IF FROM CENTRE TOWARDS RIGHT AND THEN FROM CENTRE TOWARDS LEFT.
ORIGIN=4 IF FROM CENTRE TOWARDS LEFT AND THEN FROM CENTRE TOWARDS RIGHT.
ORIGIN=5 IF FROM CENTRE TOWARDS RIGHT (SYMMETRICAL SECTION).


2     2      2.0000     3


PLEASE ENTER THE DATA FOR ELEMENT DEFINITION SEQUENCE ACCORDING TO THE GIVEN FORMAT,
ONE DEFINITION STATEMENT FOR EVERY ELEMENT ENTERED :-

SEQUENCE          ELEMENT           LENGTH OR         ANGLE OF
NUMBER (N)        TYPE (TYPE)       RADIUS            BENDING
-----------       -----------       -----------       -----------

(INTEGER)         (INTEGER)         (REAL)            (REAL)

START FROM 1,     1 = LINEAR,       POSITIVE          FOR TYPE 2
INCREMENT         2 = CIRCULAR,     LENGTH            ELEMENTS
BY 1              THE REST ARE      FOR TYPE 1,       ONLY,
UP TO 50,         ILLEGAL.          INSIDE RADIUS     ZERO FOR TYPE 1,
TO TERMINATE                        FOR TYPE 2        NON-ZERO FOR
ENTER 0.                            (MAY BE 0).       TYPE 2. *

*THE EXCEPTION BEING, WHEN N=1 AND TYPE=1 (AND ORIGIN NOT 5!),
 ANGLE OF BENDING IS TAKEN AS THE ANGLE OF INCLINATION
 BETWEEN THE HORIZONTAL-AXIS AND THE FIRST LINEAR ELEMENT,
 PERMISSIBLE RANGE IS FROM -90 TO +90 DEGREES.

NOTE THAT WHEN ORIGIN IS 3 OR 4 (DOUBLE SIDED DEFINITION),
2 SETS OF DEFINITION SEQUENCE ARE REQUIRED, ONLY 1 SET
IS REQUIRED WHEN ORIGIN IS 1,2 OR 5.
FIRST ELEMENT AND LAST ELEMENT OF THE SEQUENCE MUST BE LINEAR.
MAXIMUM NO. OF ELEMENTS IN EACH SEQUENCE MUST BE LESS THAN 50.
```

```
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
0004
```

1	1	15.00 MM.	0.00
2	2	2.00 MM.	90.00
3	1	10.00 MM.	0.00
0	0	0.00 MM.	0.00
1	1	15.00 MM.	0.00
2	2	2.00 MM.	-90.00
3	1	15.00 MM.	0.00
0	0	0.00 MM.	0.00

MEANLENGTH INFORMATION FOR INDIVIDUAL ELEMENTS:-

ELEMENT NO.	MEANLENGTH (MM.)
1	15.000
2	4.398
3	10.000
4	15.000
5	4.398
6	15.000

TOTAL MEANLENGTH= 63.796 MM.

PLEASE SELECT THE PAPERSIZE AND THE SCALE
ACCORDING TO THE GIVEN FORMAT :-

PAPERSIZE	SCALE
(INTEGER)	(REAL)

```
PAPERSIZE = 0,1,2,3,4 FOR A0,A1,A2,A3,A4 SIZES RESPECTIVELY.
SCALE= ANY POSITIVE VALUE LESS OR EQUAL TO THE FOLLOWING
       LIMITS BASED ON THE SELECTED PAPERSIZE.


FOR PAPERSIZE A0, MAXIMUM PERMISSIBLE SCALE = 23.87
FOR PAPERSIZE A1, MAXIMUM PERMISSIBLE SCALE = 16.13
FOR PAPERSIZE A2, MAXIMUM PERMISSIBLE SCALE = 10.32
FOR PAPERSIZE A3, MAXIMUM PERMISSIBLE SCALE =  6.45
FOR PAPERSIZE A4, MAXIMUM PERMISSIBLE SCALE =  3.55


PAPERSIZE A4      SCALE  1.0 : 1.0


DO YOU WANT ANY DIMENSIONING?

ENTER 1 IF YES, OR 0 IF NO.


1


DO YOU WISH TO SUPPLY ANY TITLE-BLOCK INFORMATION ?

ENTER 1 IF YES, OR 0 IF NO.


1
```

```
PLEASE MAKE INPUT SELECTION ACCORDING TO THE GIVEN LIST WITH THE REQUIRED FORMAT :-

NO.  DESCRIPTION  BLOCK  MAXIMUM NO. OF CHARACTERS
---  -----------  -----  -------------------------

 1   TITLE          2              29
 2   CUSTOMER       2              19
 3   SEC. NO.       3              14
 4   JOB NO.        3              14
 5   STRIP SIZE     3    (DETERMINED INTERNALLY)
 6   DRAWN          3               9
 7   DATE           3               9
 8   CHECKED        3               9
 9   NO.            4               9
10   MATL.          4              24
 0             (TERMINATION)

EACH SELECTION INPUT CONSISTS OF 2 LINES:-

LINE 1 SHOULD CONTAIN ITEM NO.,

LINE 2 SHOULD CONTAIN NO. OF CHARACTERS PLUS TEXT.


SELECTION NO.  1
20 CHARACTERS FOR :- ILLUSTRATION EXAMPLE

SELECTION NO.  6
 6 CHARACTERS FOR :- K.H.NG

SELECTION NO.  7
 7 CHARACTERS FOR :- 18.6.81

SELECTION NO.  0


***** END OF PROGRAM NO. 1 *****
```

```
0004
0004
008011.01.29* UAFORTRAN LOAD :EPP3136.RPLOT2Z-SV,*CR1 PLOTDATA,*TRO F9999,*LPO F2X,EXIT
0080ENTERING :MACROS.UAFORTRAN(227/)
008011.01.30*        TA AB,CM,JL
424011.01.43 0.33 CORE GIVEN 32960
00080.33 :HALTED : 94
0001DISPLAY: UAFORTRAN: GRAPHICAL OUTPUT FILE IS UACALW2381
0008DISPLAY : GINO-F MK 2.6
404011.02.30 FREE *CR1 ,232 TRANSFERS
404011.02.33 FREE *TRO ,11 TRANSFERS
404011.02.33 FREE *LPO ,28 TRANSFERS
00080.49 :DELETED : 00
404011.02.33 FREE *FH7 ,161 TRANSFERS
404011.02.33 0.49 DELETED,CLOCKED 0.14
0001DISPLAY: UAFORTRAN: NORMAL EXIT
008011.02.37* LF F2X
0004
0004
0004                                        * PLOTTING PROGRAM NO. 2 FOR ROLLED-SECTION *
0004
0004                                                --- FLOWER PATTERNS ---
0004
0004
0004
0004
0004                                                                                    SEE NOTE 1
0004                                    3                                               SEE NOTE 2
0004                                    5                                               SEE NOTE 3
0004                                    0
0004                                    1       2       -35.00      2       35.00       SEE NOTE 4A
0004                                    2       2       -60.00      2       60.00       SEE NOTE 4A
0004                                    3       2       -75.00      2       75.00       SEE NOTE 4A
0004                                    4       2       -85.00      2       85.00       SEE NOTE 4B
0004                                    5       2       -90.00      2       90.00       SEE NOTE 4A
0004                                    0       0         0.00      0        0.00       SEE NOTE 4A
0004                                   -1                                               SEE NOTE 5
0004                                    0   0   0   0
0004                                        0   0   0                                   SEE NOTE 7
0004                                  PAPERSIZE A4, MAXIMUM SCALE USED =  2.90
0004
0004
0004
0004                                  ***** END OF PROGRAM NO. 2 *****
0004
008011.02.38* UAFORTRAN LOAD :EPP3136.RMG-BK,DWNPD,DUMPON :EPP3136.RMG-BK,*CR1 PLOTDATA,*TRO F9999, DATA R9999,LINES 20000,*LPO -
    F4X,*LP1 SEC9999-R,*LP2 F4Z,EXIT
```

```
0080ENTERING :MACROS.UAFORTRAN(227/)
008011.02.39•        TA AB,CM,JL
424011.06.45 0.54 CORE GIVEN 82048
0008DISPLAY : TRACE 2 USED
@008ALLOT       *CRO     N
00080.56: MONITOR
00080.57 :HALTED : 93
0001DISPLAY: UAFORTRAN: GRAPHICAL OUTPUT FILE IS UACALN2380
0008DISPLAY : GINO-F MK 2.6
404011.10.59 FREE *LP2 ,135 TRANSFERS
404011.10.59 FREE *LP1 ,282 TRANSFERS
404011.10.59 FREE *CR1 ,232 TRANSFERS
404011.10.59 FREE *TRO ,11 TRANSFERS
404011.10.59 FREE *CRO ,45 TRANSFERS
404011.10.59 FREE *LPO ,395 TRANSFERS
00081.46 :DELETED : 00
404011.10.59 FREE *DAO ,1234 TRANSFERS
404011.11.00 FREE *FH7 ,89 TRANSFERS
404011.11.00 1.46 DELETED,CLOCKED 0.46
0001DISPLAY: UAFORTRAN: NORMAL EXIT
008011.11.01• LF F4X,TOC/INPUT FILE ENDS/
0004
0004
0004                                   * PLOTTING PROGRAM NO.4 FOR ROLLED-SECTION *
0004
0004                                              --- ROLLER PROFILES ---
0004
0004
0004
0004
0004                                                                                SEE NOTE 1
0004                      3                                                         SEE NOTE 2
0004                      5                                                         SEE NOTE 3
0004                      0                                                         SEE NOTE 4A
0004                      1      2     -35.00      2       35.00                    SEE NOTE 4A
0004                      2      2     -60.00      2       60.00                    SEE NOTE 4A
0004                      3      2     -75.00      2       75.00                    SEE NOTE 4A
0004                      4      2     -85.00      2       85.00                    SEE NOTE 4A
0004                      5      2     -90.00      2       90.00                    SEE NOTE 4A
0004                      0      0       0.00      0        0.00                    SEE NOTE 5
0004                     -1
0004                      0    0     0     0                                        SEE NOTE 7
0004                           0     0     0                                        SEE NOTE R1
0004                      1                                                         SEE NOTE R2
0004                      100.000      200.000                                      SEE NOTE R3
0004                      1                                                         SEE NOTE R4
0004                        1.000        1.000
```

```
0004                                                    1       1.000     10.000
0004                                                    -1
0004                                                     0
0004                                                        1    1    1
0004                                                        1    1    1
0004                                                        2    1    1
0004                                                        2    1    1
0004                                                        3    1    1
0004                                                        3    1    1
0004                                                        4    1    1
0004                                                        4    1    1
0004                                                        5    1    1
0004                                                        5    1    1
0004                                                        0    0    0
0004                                                    1
0004                                                    1
0004                                                          70.000     70.000
0004                                                        2    3    4    2    1
0004                                                        2    2    2    3    4
0004                                                        3    3    4    2    1
0004                                                        3    2    2    3    4
0004                                                        4    3    4    2    1
0004                                                        4    2    2    3    4
0004                                                        5    3    2    2    1
0004                                                        5    2    2    3    1
0004                                                        0    0    0    0    0
0004                                                    1
0004                                                        1    1    10.000   10.000   20.000   10.000
0004                                                        2    2    10.000   10.000   10.000   10.000
0004                                                        3    3    15.000   15.000   15.000   15.000
0004                                                        0    0     0.000    0.000    0.000    0.000
0004                                                        1    3    3    2    2
0004                                                        2    1    1    1    1
0004                                                        2    1    1    1    1
0004                                                        3    1    1    1    1
0004                                                        3    1    1    1    1
0004                                                        4    1    1    1    1
0004                                                        4    1    1    1    1
0004                                                        5    1    1    1    1
0004                                                        5    1    1    1    1
0004                                                        0    0    0    0    0
0004
0004** INPUT FILE ENDS **
0008011.11.03* ER PLOTDATA,F1X,F2X,F4X,F4Z
0008011.11.05* EJ ALLBUT,CM,0J,0L,RT(OUT9999)
>04@11.11.05 1.48 FINISHED : 1 LISTFILES
400@11.11.06 JOBTIME USED 108 ; MAXIMUM CORE USED 82048
400@11.11.06 JOB UNITS 237
```

```
SEE NOTE R5
SEE NOTE R6
SEE NOTE P1
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE P3
SEE NOTE S1
SEE NOTE S2
SEE NOTE S3
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S4
SEE NOTE S5
SEE NOTE S5
SEE NOTE S5
SEE NOTE S6
SEE NOTE S6
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
SEE NOTE S7
```

```
#LISTING OF :EPP3136.SEC9999-T(1/)  PRODUCED ON 30SEP81 AT 16.51.35

#G8.64AB AT ASTON     IN :EPP3136.MOP1 ON 12NOV81 AT 10.24.56 USING  U15

DOCUMENT     SEC9999-T


3.14159265   2   2   5
   1
    100.000    432.852    434.339     10.000
  1
  1   1      0.000    -8.179    12.866     0.830     0.000     0.000     0.000   0
  2   2     12.866     0.830    16.578     2.000    16.578    -4.472     6.472  -1
  3   1     16.578     2.000    32.285     2.000     0.000     0.000     0.000   0
  4   1     32.285     2.000    47.992     2.000     0.000     0.000     0.000   0
  5   2     47.992     2.000    50.557     2.809    47.992     6.472     4.472   1
  6   1     50.557     2.809    59.328     8.950     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  2
  1   1      1.147    -9.818    14.013    -0.809     0.000     0.000     0.000   0
  2   2     14.013    -0.809    16.578     0.000    16.578    -4.472     4.472  -1
  3   1     16.578     0.000    32.285     0.000     0.000     0.000     0.000   0
  4   1     32.285     0.000    47.992     0.000     0.000     0.000     0.000   0
  5   2     47.992     0.000    51.704     1.170    47.992     6.472     6.472   1
  6   1     51.704     1.170    60.475     7.312     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  2
    744.747   1027.375    441.651     10.000
  1
  1   1      0.000   -13.740     7.853    -0.138     0.000     0.000     0.000   0
  2   2      7.853    -0.138    11.556     2.000    11.556    -2.275     4.275  -1
  3   1     11.556     2.000    27.263     2.000     0.000     0.000     0.000   0
  4   1     27.263     2.000    42.970     2.000     0.000     0.000     0.000   0
  5   2     42.970     2.000    44.940     3.138    42.970     4.275     2.275   1
  6   1     44.940     3.138    50.294    12.410     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  2
  1   1      1.732   -14.740     9.585    -1.138     0.000     0.000     0.000   0
  2   2      9.585    -1.138    11.556     0.000    11.556    -2.275     2.275  -1
  3   1     11.556     0.000    27.263     0.000     0.000     0.000     0.000   0
  4   1     27.263     0.000    42.970     0.000     0.000     0.000     0.000   0
  5   2     42.970     0.000    46.672     2.138    42.970     4.275     4.275   1
  6   1     46.672     2.138    52.026    11.410     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  3
  1305.003   1548.256    444.148     10.000
  1
  1   1      0.000   -15.898     4.065    -0.726     0.000     0.000     0.000   0
  2   2      4.065    -0.726     7.618     2.000     7.618    -1.679     3.679  -1
  3   1      7.618     2.000    23.325     2.000     0.000     0.000     0.000   0
  4   1     23.325     2.000    39.032     2.000     0.000     0.000     0.000   0
  5   2     39.032     2.000    40.653     3.244    39.032     3.679     1.679   1
  6   1     40.653     3.244    43.425    13.586     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  2
  1   1      1.932   -16.416     5.997    -1.244     0.000     0.000     0.000   0
  2   2      5.997    -1.244     7.618     0.000     7.618    -1.679     1.679  -1
  3   1      7.618     0.000    23.325     0.000     0.000     0.000     0.000   0
  4   1     23.325     0.000    39.032     0.000     0.000     0.000     0.000   0
  5   2     39.032     0.000    42.585     2.726    39.032     3.679     3.679   1
  6   1     42.585     2.726    45.356    13.068     0.000     0.000     0.000   0
  0   0      0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
```

```
4
1798.567  2013.244   444.905     10.000
1
1    1     0.000   -16.755     1.369    -1.108     0.000     0.000     0.000     0
2    2     1.369    -1.108     4.761     2.000     4.761    -1.405     3.405    -1
3    1     4.761     2.000    20.468     2.000     0.000     0.000     0.000     0
4    1    20.468     2.000    36.174     2.000     0.000     0.000     0.000     0
5    2    36.174     2.000    37.574     3.282    36.174     3.405     1.405     1
6    1    37.574     3.282    38.507    13.948     0.000     0.000     0.000     0
0    0     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0
2
1    1     1.992   -16.929     3.361    -1.282     0.000     0.000     0.000     0
2    2     3.361    -1.282     4.761     0.000     4.761    -1.405     1.405    -1
3    1     4.761     0.000    20.468     0.000     0.000     0.000     0.000     0
4    1    20.468     0.000    36.174     0.000     0.000     0.000     0.000     0
5    2    36.174     0.000    39.566     3.108    36.174     3.405     3.405     1
6    1    39.566     3.108    40.500    13.774     0.000     0.000     0.000     0
0    0     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0
5
2243.563  2443.560   445.000     10.000
1
1    1     0.000   -17.000     0.000    -2.000     0.000     0.000     0.000     0
2    2     0.000    -2.000     4.000     2.000     4.000    -2.000     4.000    -1
3    1     4.000     2.000    19.000     2.000     0.000     0.000     0.000     0
4    1    19.000     2.000    34.000     2.000     0.000     0.000     0.000     0
5    2    34.000     2.000    36.000     4.000    34.000     4.000     2.000     1
6    1    36.000     4.000    36.000    14.000     0.000     0.000     0.000     0
0    0     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0
2
1    1     2.000   -17.000     2.000    -2.000     0.000     0.000     0.000     0
2    2     2.000    -2.000     4.000     0.000     4.000    -2.000     2.000    -1
3    1     4.000     0.000    19.000     0.000     0.000     0.000     0.000     0
4    1    19.000     0.000    34.000     0.000     0.000     0.000     0.000     0
5    2    34.000     0.000    38.000     4.000    34.000     4.000     4.000     1
6    1    38.000     4.000    38.000    14.000     0.000     0.000     0.000     0
0    0     0.000     0.000     0.000     0.000     0.000     0.000     0.000     0
```

#LISTING OF :EPP3136.SEC9999-R(1/) PRODUCED ON 27OCT81 AT 11.10.59

#68.64AB AT ASTON IN :EPP3136.MOP1* ON 12NOV81 AT 10.24.36 USING U15

DOCUMENT SEC9999-R

```
3.14159265   2   2   5
   1   100.000   200.000    70.000    70.000
   442.228   113.219   120.000     1.000   0   0
 1
   1   1     0.000  -32.193   15.000  -32.193    0.000   0.000   0.000   0
   2   1    15.000  -32.193   15.000  -17.193    0.000   0.000   0.000   0
   3   1    15.000  -17.193   30.000  -17.193    0.000   0.000   0.000   0
   4   1    30.000  -17.193   42.287   -8.589    0.000   0.000   0.000   0
   5   1    42.287   -8.589   55.973    0.993    0.000   0.000   0.000   0
   6   2    55.973    0.993   59.800    2.200   59.800  -4.472   6.672  -1
   7   1    59.800    2.200   59.800    2.193    0.000   0.000   0.000   0
   8   1    59.800    2.193   75.506    2.193    0.000   0.000   0.000   0
   9   1    75.506    2.193   91.213    2.193    0.000   0.000   0.000   0
  10   1    91.213    2.193   91.213    2.200    0.000   0.000   0.000   0
  11   2    91.213    2.200   93.663    2.973   91.213   6.472   4.272   1
  12   1    93.663    2.973  103.253    9.687    0.000   0.000   0.000   0
  13   1   103.253    9.687  104.515    7.885    0.000   0.000   0.000   0
  14   1   104.515    7.885  112.707   13.621    0.000   0.000   0.000   0
  15   1   112.707   13.621  122.707   13.621    0.000   0.000   0.000   0
  16   1   122.707   13.621  122.707   23.621    0.000   0.000   0.000   0
  17   1   122.707   23.621  132.707   23.621    0.000   0.000   0.000   0
   0   0     0.000    0.000    0.000    0.000    0.000   0.000   0.000   0
 2
   1   1     0.000  -32.193   15.000  -32.193    0.000   0.000   0.000   0
   2   1    15.000  -32.193   15.000  -17.193    0.000   0.000   0.000   0
   3   1    15.000  -17.193   30.000  -17.193    0.000   0.000   0.000   0
   4   1    30.000  -17.193   42.287   -8.589    0.000   0.000   0.000   0
   5   1    42.287   -8.589   43.549  -10.391    0.000   0.000   0.000   0
   6   1    43.549  -10.391   57.235   -0.809    0.000   0.000   0.000   0
   7   2    57.235   -0.809   59.800    0.000   59.800  -4.472   4.472  -1
   8   1    59.800    0.000   59.800    0.000    0.000   0.000   0.000   0
   9   1    59.800    0.000   75.506    0.000    0.000   0.000   0.000   0
  10   1    75.506    0.000   91.213    0.000    0.000   0.000   0.000   0
  11   1    91.213    0.000   91.213    0.000    0.000   0.000   0.000   0
  12   2    91.213    0.000   94.925    1.170   91.213   6.472   6.472   1
  13   1    94.925    1.170  104.515    7.885    0.000   0.000   0.000   0
  14   1   104.515    7.885  112.707   13.621    0.000   0.000   0.000   0
  15   1   112.707   13.621  122.707   13.621    0.000   0.000   0.000   0
  16   1   122.707   13.621  122.707   23.621    0.000   0.000   0.000   0
  17   1   122.707   23.621  132.707   23.621    0.000   0.000   0.000   0
   0   0     0.000    0.000    0.000    0.000    0.000   0.000   0.000   0
   2   100.000   200.000    70.000    70.000
   675.427   107.936   120.000     1.000   1   1
 1
   1   1     0.000   27.200    0.000   17.200    0.000   0.000   0.000   0
   2   1     0.000   17.200   20.000   17.200    0.000   0.000   0.000   0
   3   1    20.000   17.200   20.000    7.200    0.000   0.000   0.000   0
   4   1    20.000    7.200   28.660    2.200    0.000   0.000   0.000   0
   5   1    28.660    2.200   28.660    2.193    0.000   0.000   0.000   0
   6   1    28.660    2.193   44.367    2.193    0.000   0.000   0.000   0
   7   1    44.367    2.193   60.074    2.193    0.000   0.000   0.000   0
   8   1    60.074    2.193   60.074    2.200    0.000   0.000   0.000   0
   9   2    60.074    2.200   61.871    3.238   60.074   4.275   2.075   1
  10   1    61.871    3.238   67.725   13.376    0.000   0.000   0.000   0
  11   1    67.725   13.376   76.385   18.376    0.000   0.000   0.000   0
  12   1    76.385   18.376   76.385   28.376    0.000   0.000   0.000   0
  13   1    76.385   28.376   96.385   28.376    0.000   0.000   0.000   0
  14   1    96.385   28.376   96.385   38.376    0.000   0.000   0.000   0
   0   0     0.000    0.000    0.000    0.000    0.000   0.000   0.000   0
```

```
2
  1   1     0.000   -40.606     0.000   -30.606    0.000    0.000    0.000   0
  2   1     0.000   -30.606    20.000   -30.606    0.000    0.000    0.000   0
  3   1    20.000   -30.606    20.000   -20.606    0.000    0.000    0.000   0
  4   1    20.000   -20.606    28.660   -15.606    0.000    0.000    0.000   0
  5   1    28.660   -15.606    37.014    -1.138    0.000    0.000    0.000   0
  6   2    37.014    -1.138    38.984     0.000   38.984   -2.275    2.275  -1
  7   1    38.984     0.000    38.984     0.000    0.000    0.000    0.000   0
  8   1    38.984     0.000    54.691     0.000    0.000    0.000    0.000   0
  9   1    54.691     0.000    70.398     0.000    0.000    0.000    0.000   0
 10   1    70.398     0.000    70.398     0.000    0.000    0.000    0.000   0
 11   1    70.398     0.000    79.058    -5.000    0.000    0.000    0.000   0
 12   1    79.058    -5.000    79.058   -15.000    0.000    0.000    0.000   0
 13   1    79.058   -15.000    99.058   -15.000    0.000    0.000    0.000   0
 14   1    99.058   -15.000    99.058   -25.000    0.000    0.000    0.000   0
  0   0     0.000     0.000     0.000     0.000    0.000    0.000    0.000   0
3
  1   1     0.000   -15.309    10.000   -15.309    0.000    0.000    0.000   0
  2   1    10.000   -15.309    10.000   -35.309    0.000    0.000    0.000   0
  3   1    10.000   -35.309    20.000   -35.309    0.000    0.000    0.000   0
  4   1    20.000   -35.309    25.000   -43.969    0.000    0.000    0.000   0
  5   1    25.000   -43.969    26.100   -42.064    0.000    0.000    0.000   0
  6   1    26.100   -42.064    40.569   -50.417    0.000    0.000    0.000   0
  7   2    40.569   -50.417    42.806   -54.293   38.331  -54.293    4.475  -1
  8   1    42.806   -54.293    47.806   -45.633    0.000    0.000    0.000   0
  9   1    47.806   -45.633    57.806   -45.633    0.000    0.000    0.000   0
 10   1    57.806   -45.633    57.806   -25.633    0.000    0.000    0.000   0
 11   1    57.806   -25.633    67.806   -25.633    0.000    0.000    0.000   0
  0   0     0.000     0.000     0.000     0.000    0.000    0.000    0.000   0
4
  1   1     0.000    25.633    10.000    25.633    0.000    0.000    0.000   0
  2   1    10.000    25.633    10.000    45.633    0.000    0.000    0.000   0
  3   1    10.000    45.633    20.000    45.633    0.000    0.000    0.000   0
  4   1    20.000    45.633    25.000    54.293    0.000    0.000    0.000   0
  5   2    25.000    54.293    27.138    50.591   29.275   54.293    4.275   1
  6   1    27.138    50.591    37.276    44.737    0.000    0.000    0.000   0
  7   1    37.276    44.737    38.376    46.642    0.000    0.000    0.000   0
  8   1    38.376    46.642    43.376    37.982    0.000    0.000    0.000   0
  9   1    43.376    37.982    53.376    37.982    0.000    0.000    0.000   0
 10   1    53.376    37.982    53.376    17.982    0.000    0.000    0.000   0
 11   1    53.376    17.982    63.376    17.982    0.000    0.000    0.000   0
  0   0     0.000     0.000     0.000     0.000    0.000    0.000    0.000   0
  3   100.000   200.000    70.000    70.000
  901.494   103.777   120.000     1.000   1   1
1
  1   1     0.000    27.200     0.000    17.200    0.000    0.000    0.000   0
  2   1     0.000    17.200    20.000    17.200    0.000    0.000    0.000   0
  3   1    20.000    17.200    20.000     7.200    0.000    0.000    0.000   0
  4   1    20.000     7.200    28.660     2.200    0.000    0.000    0.000   0
  5   1    28.660     2.200    28.660     2.193    0.000    0.000    0.000   0
  6   1    28.660     2.193    44.367     2.193    0.000    0.000    0.000   0
  7   1    44.367     2.193    60.074     2.193    0.000    0.000    0.000   0
  8   1    60.074     2.193    60.074     2.200    0.000    0.000    0.000   0
  9   2    60.074     2.200    61.502     3.296   60.074    3.679    1.479   1
 10   1    61.502     3.296    64.532    14.604    0.000    0.000    0.000   0
 11   1    64.532    14.604    73.192    19.604    0.000    0.000    0.000   0
 12   1    73.192    19.604    73.192    29.604    0.000    0.000    0.000   0
 13   1    73.192    29.604    93.192    29.604    0.000    0.000    0.000   0
 14   1    93.192    29.604    93.192    39.604    0.000    0.000    0.000   0
  0   0     0.000     0.000     0.000     0.000    0.000    0.000    0.000   0
2
  1   1     0.000   -42.382     0.000   -32.382    0.000    0.000    0.000   0
  2   1     0.000   -32.382    20.000   -32.382    0.000    0.000    0.000   0
  3   1    20.000   -32.382    20.000   -22.382    0.000    0.000    0.000   0
  4   1    20.000   -22.382    28.660   -17.382    0.000    0.000    0.000   0
  5   1    28.660   -17.382    32.984    -1.244    0.000    0.000    0.000   0
  6   2    32.984    -1.244    34.606     0.000   34.606   -1.679    1.679  -1
  7   1    34.606     0.000    34.606     0.000    0.000    0.000    0.000   0
  8   1    34.606     0.000    50.312     0.000    0.000    0.000    0.000   0
  9   1    50.312     0.000    66.019     0.000    0.000    0.000    0.000   0
 10   1    66.019     0.000    66.019     0.000    0.000    0.000    0.000   0
 11   1    66.019     0.000    74.680    -5.000    0.000    0.000    0.000   0
 12   1    74.680    -5.000    74.680   -15.000    0.000    0.000    0.000   0
 13   1    74.680   -15.000    94.680   -15.000    0.000    0.000    0.000   0
 14   1    94.680   -15.000    94.680   -25.000    0.000    0.000    0.000   0
  0   0     0.000     0.000     0.000     0.000    0.000    0.000    0.000   0
```

```
3
  1   1     0.000   -19.688    10.000   -19.688     0.000     0.000     0.000   0
  2   1    10.000   -19.688    10.000   -39.688     0.000     0.000     0.000   0
  3   1    10.000   -39.688    20.000   -39.688     0.000     0.000     0.000   0
  4   1    20.000   -39.688    25.000   -48.348     0.000     0.000     0.000   0
  5   1    25.000   -48.348    25.569   -46.223     0.000     0.000     0.000   0
  6   1    25.569   -46.223    41.707   -50.547     0.000     0.000     0.000   0
  7   2    41.707   -50.547    44.582   -54.293    40.703   -54.293     3.879  -1
  8   1    44.582   -54.293    49.582   -45.633     0.000     0.000     0.000   0
  9   1    49.582   -45.633    59.582   -45.633     0.000     0.000     0.000   0
 10   1    59.582   -45.633    59.582   -25.633     0.000     0.000     0.000   0
 11   1    59.582   -25.633    69.582   -25.633     0.000     0.000     0.000   0
  0   0     0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
4
  1   1     0.000    25.633    10.000    25.633     0.000     0.000     0.000   0
  2   1    10.000    25.633    10.000    45.633     0.000     0.000     0.000   0
  3   1    10.000    45.633    20.000    45.633     0.000     0.000     0.000   0
  4   1    20.000    45.633    25.000    54.293     0.000     0.000     0.000   0
  5   2    25.000    54.293    27.726    50.740    28.679    54.293     3.679   1
  6   1    27.726    50.740    39.034    47.710     0.000     0.000     0.000   0
  7   1    39.034    47.710    39.604    49.835     0.000     0.000     0.000   0
  8   1    39.604    49.835    44.604    41.175     0.000     0.000     0.000   0
  9   1    44.604    41.175    54.604    41.175     0.000     0.000     0.000   0
 10   1    54.604    41.175    54.604    21.175     0.000     0.000     0.000   0
 11   1    54.604    21.175    64.604    21.175     0.000     0.000     0.000   0
  0   0     0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
  4   100.000   200.000    70.000    70.000
 1122.367   100.754   120.000     1.000   1   1
1
  1   1     0.000    27.200     0.000    17.200     0.000     0.000     0.000   0
  2   1     0.000    17.200    20.000    17.200     0.000     0.000     0.000   0
  3   1    20.000    17.200    20.000     7.200     0.000     0.000     0.000   0
  4   1    20.000     7.200    28.660     2.200     0.000     0.000     0.000   0
  5   1    28.660     2.200    28.660     2.193     0.000     0.000     0.000   0
  6   1    28.660     2.193    44.367     2.193     0.000     0.000     0.000   0
  7   1    44.367     2.193    60.074     2.193     0.000     0.000     0.000   0
  8   1    60.074     2.193    60.074     2.200     0.000     0.000     0.000   0
  9   2    60.074     2.200    61.274     3.300    60.074     3.405    -1.205  -1
 10   1    61.274     3.300    62.294    14.962     0.000     0.000     0.000   0
 11   1    62.294    14.962    70.955    19.962     0.000     0.000     0.000   0
 12   1    70.955    19.962    70.955    29.962     0.000     0.000     0.000   0
 13   1    70.955    29.962    90.955    29.962     0.000     0.000     0.000   0
 14   1    90.955    29.962    90.955    39.962     0.000     0.000     0.000   0
  0   0     0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
2
  1   1     0.000   -42.926     0.000   -32.926     0.000     0.000     0.000   0
  2   1     0.000   -32.926    20.000   -32.926     0.000     0.000     0.000   0
  3   1    20.000   -32.926    20.000   -22.926     0.000     0.000     0.000   0
  4   1    20.000   -22.926    28.660   -17.926     0.000     0.000     0.000   0
  5   1    28.660   -17.926    30.116    -1.282     0.000     0.000     0.000   0
  6   2    30.116    -1.282    31.516     0.000    31.516    -1.405     1.405  -1
  7   1    31.516     0.000    31.516     0.000     0.000     0.000     0.000   0
  8   1    31.516     0.000    47.223     0.000     0.000     0.000     0.000   0
  9   1    47.223     0.000    62.929     0.000     0.000     0.000     0.000   0
 10   1    62.929     0.000    62.929     0.000     0.000     0.000     0.000   0
 11   1    62.929     0.000    71.590    -5.000     0.000     0.000     0.000   0
 12   1    71.590    -5.000    71.590   -15.000     0.000     0.000     0.000   0
 13   1    71.590   -15.000    91.590   -15.000     0.000     0.000     0.000   0
 14   1    91.590   -15.000    91.590   -25.000     0.000     0.000     0.000   0
  0   0     0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
3
  1   1     0.000   -22.777    10.000   -22.777     0.000     0.000     0.000   0
  2   1    10.000   -22.777    10.000   -42.777     0.000     0.000     0.000   0
  3   1    10.000   -42.777    20.000   -42.777     0.000     0.000     0.000   0
  4   1    20.000   -42.777    25.000   -51.438     0.000     0.000     0.000   0
  5   1    25.000   -51.438    25.192   -49.246     0.000     0.000     0.000   0
  6   1    25.192   -49.246    41.835   -50.702     0.000     0.000     0.000   0
  7   2    41.835   -50.702    45.126   -54.293    41.521   -54.293     3.605  -1
  8   1    45.126   -54.293    50.126   -45.633     0.000     0.000     0.000   0
  9   1    50.126   -45.633    60.126   -45.633     0.000     0.000     0.000   0
 10   1    60.126   -45.633    60.126   -25.633     0.000     0.000     0.000   0
 11   1    60.126   -25.633    70.126   -25.633     0.000     0.000     0.000   0
  0   0     0.000     0.000     0.000     0.000     0.000     0.000     0.000   0
```

```
4
  1   1     0.000   25.633   10.000   25.633    0.000    0.000    0.000  0
  2   1    10.000   25.633   10.000   45.633    0.000    0.000    0.000  0
  3   1    10.000   45.633   20.000   45.633    0.000    0.000    0.000  0
  4   1    20.000   45.633   25.000   54.293    0.000    0.000    0.000  0
  5   2    25.000   54.293   28.108   50.901   28.405   54.293    3.405  1
  6   1    28.108   50.901   39.770   49.881    0.000    0.000    0.000  0
  7   1    39.770   49.881   39.962   52.073    0.000    0.000    0.000  0
  8   1    39.962   52.073   44.962   43.412    0.000    0.000    0.000  0
  9   1    44.962   43.412   54.962   43.412    0.000    0.000    0.000  0
 10   1    54.962   43.412   54.962   23.412    0.000    0.000    0.000  0
 11   1    54.962   23.412   64.962   23.412    0.000    0.000    0.000  0
  0   0     0.000    0.000    0.000    0.000    0.000    0.000    0.000  0
  5   100.000   200.000    70.000    70.000
1340.566     99.200    120.000     1.000   1   1
1
  1   1     0.000   27.200    0.000   17.200    0.000    0.000    0.000  0
  2   1     0.000   17.200   20.000   17.200    0.000    0.000    0.000  0
  3   1    20.000   17.200   20.000    7.200    0.000    0.000    0.000  0
  4   1    20.000    7.200   28.660    2.200    0.000    0.000    0.000  0
  5   1    28.660    2.200   28.660    2.193    0.000    0.000    0.000  0
  6   1    28.660    2.193   43.660    2.193    0.000    0.000    0.000  0
  7   1    43.660    2.193   58.660    2.193    0.000    0.000    0.000  0
  8   1    58.660    2.193   58.660    2.200    0.000    0.000    0.000  0
  9   2    58.660    2.200   60.460    4.000   58.660    4.000    1.800  1
 10   1    60.460    4.000   60.460   15.000    0.000    0.000    0.000  0
 11   1    60.460   15.000   62.660   15.000    0.000    0.000    0.000  0
 12   1    62.660   15.000   71.320   20.000    0.000    0.000    0.000  0
 13   1    71.320   20.000   71.320   30.000    0.000    0.000    0.000  0
 14   1    71.320   30.000   91.320   30.000    0.000    0.000    0.000  0
 15   1    91.320   30.000   91.320   40.000    0.000    0.000    0.000  0
  0   0     0.000    0.000    0.000    0.000    0.000    0.000    0.000  0
2
  1   1     0.000  -43.000    0.000  -33.000    0.000    0.000    0.000  0
  2   1     0.000  -33.000   20.000  -33.000    0.000    0.000    0.000  0
  3   1    20.000  -33.000   20.000  -23.000    0.000    0.000    0.000  0
  4   1    20.000  -23.000   28.660  -18.000    0.000    0.000    0.000  0
  5   1    28.660  -18.000   30.860  -18.000    0.000    0.000    0.000  0
  6   1    30.860  -18.000   30.860   -2.000    0.000    0.000    0.000  0
  7   2    30.860   -2.000   32.860    0.000   32.860   -2.000    2.000 -1
  8   1    32.860    0.000   32.860    0.000    0.000    0.000    0.000  0
  9   1    32.860    0.000   47.860    0.000    0.000    0.000    0.000  0
 10   1    47.860    0.000   62.860    0.000    0.000    0.000    0.000  0
 11   1    62.860    0.000   62.860    0.000    0.000    0.000    0.000  0
 12   1    62.860    0.000   71.520   -5.000    0.000    0.000    0.000  0
 13   1    71.520   -5.000   71.520  -15.000    0.000    0.000    0.000  0
 14   1    71.520  -15.000   91.520  -15.000    0.000    0.000    0.000  0
 15   1    91.520  -15.000   91.520  -25.000    0.000    0.000    0.000  0
  0   0     0.000    0.000    0.000    0.000    0.000    0.000    0.000  0
3
  1   1     0.000  -22.140   10.000  -22.140    0.000    0.000    0.000  0
  2   1    10.000  -22.140   10.000  -42.140    0.000    0.000    0.000  0
  3   1    10.000  -42.140   20.000  -42.140    0.000    0.000    0.000  0
  4   1    20.000  -42.140   25.000  -50.800    0.000    0.000    0.000  0
  5   1    25.000  -50.800   41.000  -50.800    0.000    0.000    0.000  0
  6   2    41.000  -50.800   45.200  -55.000   41.000  -55.000    4.200 -1
  7   1    45.200  -55.000   50.200  -46.340    0.000    0.000    0.000  0
  8   1    50.200  -46.340   60.200  -46.340    0.000    0.000    0.000  0
  9   1    60.200  -46.340   60.200  -26.340    0.000    0.000    0.000  0
 10   1    60.200  -26.340   70.200  -26.340    0.000    0.000    0.000  0
  0   0     0.000    0.000    0.000    0.000    0.000    0.000    0.000  0
4
  1   1     0.000   26.340   10.000   26.340    0.000    0.000    0.000  0
  2   1    10.000   26.340   10.000   46.340    0.000    0.000    0.000  0
  3   1    10.000   46.340   20.000   46.340    0.000    0.000    0.000  0
  4   1    20.000   46.340   25.000   55.000    0.000    0.000    0.000  0
  5   2    25.000   55.000   29.000   51.000   29.000   55.000    4.000  1
  6   1    29.000   51.000   40.000   51.000    0.000    0.000    0.000  0
  7   1    40.000   51.000   45.000   42.340    0.000    0.000    0.000  0
  8   1    45.000   42.340   55.000   42.340    0.000    0.000    0.000  0
  9   1    55.000   42.340   55.000   22.340    0.000    0.000    0.000  0
 10   1    55.000   22.340   65.000   22.340    0.000    0.000    0.000  0
  0   0     0.000    0.000    0.000    0.000    0.000    0.000    0.000  0
```

APPENDIX 2

SYMBOLIC MAJOR LOGIC REPRESENTATION

The Symbolic Major Logic Representation (SMLR) notations are specially designed and used for the purpose of summarizing the major logic patterns of the software highlighting the key relationship between one module and another during execution. The notations adopted are as follows:

1. Entry Item

Format: (NAME

The open-parenthesis indicates that execution control is passed to the module known as NAME.

2. Exit Item

Format: NAME)

The closed-parenthesis indicates that execution control is passed from the module known as name to whatever that follows. A pair of Entry Item and Exit Item of the same name forms a module.

3. Series Execution

Symbol: -

(dash)

This symbol follows either an Entry Item or an Exit Item to indicate that execution continues from that point to whatever that follows.

4. Parallel Execution

Symbol: /

(slash)

This symbol follows an Exit Item to indicate that the preceding module is mutually exclusive to the succeeding module in execution. Double slashes are used to indicate that the preceding module may be bypassed during execution and control is then passed on to whatever that follows.

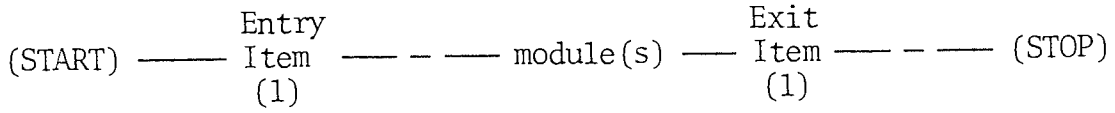
5. Looping

Format: < module(s) >

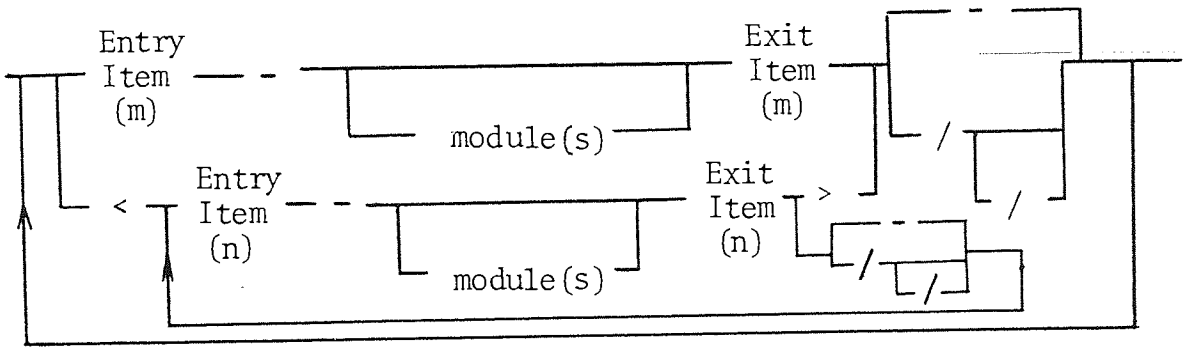
The looping symbols, which are in the form of a pair of pointed brackets, indicate that the execution of the module(s) enclosed by them is repetitive.

The syntax diagrams for constructing the required software logic patterns using the SMLR notations are as follows:

1. Overall Structure

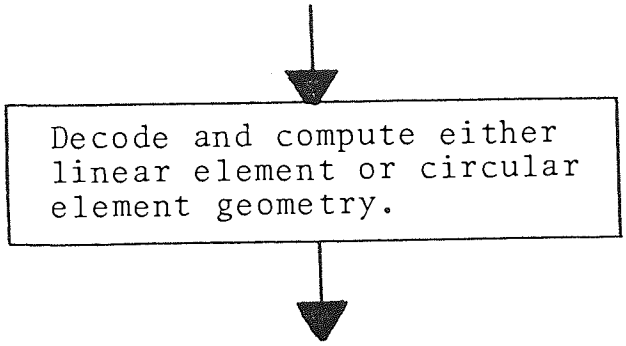


2. Module Structure



(where both m and n are unique integers greater than 1)

The notations may be used together with flow-charts to provide concise documentation for the software, as illustrated by the following example:



	ENTRY	EXIT
OP 1	(DCODE- (STL1- (ARC1-	STL1)/ ARC1)- DCODE)-

The notations may also be used independently as a short-hand for logic representation, as illustrated by the following example:

..... (DCODE- (STL1- STL1)/(ARC1- ARC1)- DCODE)- .....

If in addition, comments are required, they may be inserted randomly at appropriate points as self-contained and recognizable units enclosed by quotation marks (or other suitable symbols), as illustrated by the following example:

..... (DCODE- "Input decoding" (STL1- "Linear element processing" STL1)/(ARC1- "Circular element processing" ARC1)- DCODE)- .....

The normal techniques of spacing, indentation and paragraphing for clearer presentation may also be used

without affecting the logic. The possibility of optionally nesting module(s) within module(s) permits an unlimited amount of details to be included or excluded at will.

APPENDIX 3

SYSTEM INFORMATION

1. FILE SYSTEM-MACRO

Job processing instructions to the computer, as retained in the various macro files, corresponding to each of the existing processing commands, namely the SECTION, FLOWER, TEMPLATE and ROLL commands.

2. FILE F-INPUT-INFO

Explanatory notes for input data items in the flower pattern input data file.

3. FILE R-INPUT-INFO

Explanatory notes for input data items in the roll input data file.

```
#LISTING OF :EPP3136.SYSTEM-MACRO(1/)   PRODUCED ON 30OCT81 AT 17.05.10
#G8.64AB AT ASTON    IN :EPP3136.MOP1° ON 12NOV81 AT 10.25.04 USING  U15
DOCUMENT    SYSTEM-MACRO


SYSTEM MACRO FILES FOR PROCESSING COMMANDS ON THE ICL1904S
==========================================================

SECTION
-------

OBEY :EPP3136.MACSECT,PARAM(%A,%B,%C)

FLOWER
------

OBEY :EPP3136.MACFLOWER,PARAM(%A,%B,%C,%D)

TEMPLATE
--------

OBEY :EPP3136.MACTEMPLATE,PARAM(%A,%B,%C,%D)

ROLL
----

OBEY :EPP3136.MACROLL,PARAM(%A,%B,%C,%D,%E)

(ALL THE ABOVE FILES MUST BE CREATED IN THE USERS° OWN DIRECTORY.
 THE FOLLOWING FILES ARE IN THE DIRECTORY OF USER-NUMBER :EPP3136.)

MACSECT
-------

RJ MOP%A,:EPP3136.ROLLSECT,PARAM(%A,%B,%C),JD(JT300)

MACFLOWER
---------

RJ MOP%A,:EPP3136.ROLLFLOWER,PARAM(%A,%B,%C,%D),JD(JT300,MZ35K)

MACTEMP
-------

RJ MOP%A,:EPP3136.ROLLTEMP,PARAM(%A,%B,%C,%D),JD(JT300,MZ35K)

MACROLL
-------

RJ MOP%A,:EPP3136.ROLLDRAW,PARAM(%A,%B,%C,%D,%E),JD(JT600,MZ86K)
```

```
ROLLSECT
--------

LF %B,NU
UAFORTRAN LOAD :EPP3136.RPLOT1Z-SV,DATA %B,*LP1 PLOTDATA,-
LINES 10000,*LP0 F1X,EXIT
LF F1X
ER PLOTDATA,F1X
EJ ALLBUT,CM,OJ,OL,RT(%C)


ROLLFLOWER
----------

LF %B,NU
LF %C,NU
UAFORTRAN LOAD :EPP3136.RPLOT1Z-SV,DATA %B,*LP1 PLOTDATA,-
LINES 10000,*LP0 F1X,EXIT
LF F1X
UAFORTRAN LOAD :EPP3136.RPLOT2Z-SV,*CR1 PLOTDATA,*TR0 %C,-
*LP0 F2X,EXIT
LF F2X
ER PLOTDATA,F1X,F2X
EJ ALLBUT,CM,OJ,OL,RT(%D)


ROLLTEMP
--------

LF %B,NU
LF %C,NU
UAFORTRAN LOAD :EPP3136.RPLOT1Z-SV,DATA %B,*LP1 PLOTDATA,-
LINES 10000,*LP0 F1X,EXIT
LF F1X
UAFORTRAN LOAD :EPP3136.RPLOT2Z-SV,*CR1 PLOTDATA,*TR0 %C,-
*LP0 F2X,EXIT
LF F2X
UAFORTRAN LOAD :EPP3136.RPLOT3Z-SV,*CR1 PLOTDATA,*TR0 %C,-
*LP1 %A-T,*LP0 F3X,LINES 10000,EXIT
ER PLOTDATA,F1X,F2X,F3X
EJ ALLBUT,CM,OJ,OL,RT(%D)


ROLLDRAW
--------

LF %B,NU
LF %C,NU
LF %D,NU
UAFORTRAN LOAD :EPP3136.RPLOT1Z-SV,DATA %B,*LP1 PLOTDATA,-
LINES 10000,*LP0 F1X,EXIT
LF F1X
UAFORTRAN LOAD :EPP3136.RPLOT2Z-SV,*CR1 PLOTDATA,*TR0 %C,-
*LP0 F2X,EXIT
LF F2X
UAFORTRAN LOAD :EPP3136.RMQ-BK,OWNPD,DUMPON :EPP3136.RMQ-BK,*CR1 PLOTDATA,-
*TR0 %C, DATA %D,LINES 20000,*LP0 F4X,*LP1 %A-R,*LP2 F4Z,EXIT
LF F4X,TOC/INPUT FILE ENDS/
ER PLOTDATA,F1X,F2X,F4X,F4Z
EJ ALLBUT,CM,OJ,OL,RT(%E)
```

```
#LISTING OF :EPP3136.F-INPUT-INFO(1/)   PRODUCED ON 27OCT81 AT 10.32.29

#G8.64AB AT ASTON    IN :EPP3136.MOP1' ON 12NOV81 AT 10.25.12 USING  U15

DOCUMENT     F-INPUT-INFO

C-----------------------------------------------------------------------
C
C      *** INPUT DATA FORMAT (FLOWER PATTERNS) ***
C
C-----------------------------------------------------------------------
C
C   1. JRUN = 1 (IF FLOWER PATTERNS ONLY), OR
C             2 (IF ROLLER-PLOTTINGS ONLY), OR
C             3 (IF FLOWER-PATTERNS AND ROLLER-PLOTTINGS).
C
C   2. NSTAGE = TOTAL NUMBER OF STAGES
C               (THE PERMISSIBLE RANGE IS FROM 1 TO 50)
C
C   3. JBEND = SELECTION OF CIRCULAR-ELEMENT BENDING DEFINITION OPTION:-
C              0 (IF SIMPLE ELEMENT DEFINITION), OR
C              1 (IF COMPOSITE PERCENTAGE ELEMENT DEFINITION).
C
C   4A. ISEQ  = BENDING STAGE SEQUENCE NUMBER (LARGEST OF WHICH = NSTAGE).
C       IELML = LEFT-HAND ELEMENT (CIRCULAR) TO BE BENT
C       XANGL = CUMULATIVE ANGLE OF BENDING FOR THE LEFT-HAND ELEMENT.
C       IELMR = RIGHT-HAND ELEMENT (CIRCULAR) TO BE BENT.
C       XANGR = CUMULATIVE ANGLE OF BENDING FOR THE RIGHT-HAND ELEMENT.
C
C   4B. ** THIS IS THE COMPOSITE PERCENTAGE ELEMENT DEFINITION OPTION **
C
C       ISEQ  = (AS IN NOTE 4A)
C       IELML = (AS IN NOTE 4A)
C       XANGL = (AS IN NOTE 4A)
C       IPCL1 = PERCENTAGE LENGTH OF LEADING LINEAR PART.
C       IPCL2 = PERCENTAGE LENGTH OF CIRCULAR PART.
C       IPCL3 = PERCENTAGE LENGTH OF TRAILING LINEAR PART.
C       IELMR = (AS IN NOTE 4A)
C       XANGR = (AS IN NOTE 4A)
C       IPCR1 = PERCENTAGE LENGTH OF LEADING LINEAR PART.
C       IPCR2 = PERCENTAGE LENGTH OF CIRCULAR PART.
C       IPCR3 = PERCENTAGE LENGTH OF TRAILING LINEAR PART.
C
C       (NOTE THAT THE IPCL'S AND THE IPCR'S SHOULD BE INTEGER VALUES
C        IN THE RANGE 0 TO 100 AND THE SUM SHOULD BE 100 EXACTLY IN
C        EACH CASE).
C
C       N.B. :- TO TERMINATE THE DEFINITION SEQUENCE FOR EITHER 4A OR 4B,
C               JUST ENTER 0 FOR ISEQ AND DUMMY VALUES FOR THE REST.
C   5. ** THIS IS THE RADII-SHARPENING OPTION **
C
C      JSHARP = 1 (IF STAGE NO. ARE TO BE ENTERED INDIVIDUALLY), OR
C              -1 (IF ALL STAGES EXCEPT LAST STAGE REQUIRE SHARPENING
C                  OF RADII), OR
C               0 (IF RADII-SHARPENING IS NOT REQUIRED AT ALL).
C      N.B. :- IF JSHARP IS 0 OR -1, NO FURTHER INPUT DATA IS REQUIRED
C              FOR THIS OPTION, PROCEED TO THE NEXT OPTION INPUT.
C
C      ISQSH = STAGE NO. WITH RADII-SHARPENING ACTIVE.
C
C   6. ** THIS IS THE FIXED PERCENTAGE COMPOSITE LENGTH OPTION **
C
C      ISQCP = ENTRY NO. WITH FIXED PERCENTAGE COMPOSITE LENGTHS
C      IECL1 = L.H.S. ELEMENT NO. WITH SUCH COMPOSITION.
C      IECL2 = STAGE NO. WHEN IT STARTS TO BE ACTIVE.
C      IECL3 = STAGE NO. WHEN IT CEASES TO BE ACTIVE.
C      IECR1 = R.H.S. ELEMENT NO. WITH SUCH COMPOSITION.
C      IECR2 = STAGE NO. WHEN IT STARTS TO BE ACTIVE.
C      IECR3 = STAGE NO. WHEN IT CEASES TO BE ACTIVE.
C
C      N.B. :- IF ONLY ELEMENT ON ONE-SIDE IS DESIRED, THEN ENTER 0 FOR
C              ELEMENT NO. ON THE IRRELEVANT SIDE.
C              TO TERMINATE THIS SEQUENCE, ENTER 0 FOR ALL VALUES IN A LINE.
C
C
C-----------------------------------------------------------------------
```

```
#LISTING OF :EPP3136.R-INPUT-INFO(1/)    PRODUCED ON 11NOV81 AT 14.17.55

#G8.64AB AT ASTON     IN :EPP3136.MOP1* ON 12NOV81 AT 10.25.18 USING  U15

DOCUMENT     R-INPUT-INFO

C-----------------------------------------------------------------------
C
C      *** INPUT DATA FORMAT (ROLL DESIGNS) ***
C
C-----------------------------------------------------------------------
C
C  R1. IPASHT = +1 (IF CONSISTENT PASS-HEIGHT & C-TO-C DISTANCE),OR
C               -1 (IF INCONSISTENT PASS-HEIGHTS & C-TO-C DISTANCES)
C
C  R2. PASHT = PASS-HEIGHT (OR RADIUS OF THE BOTTOM-ROLL)
C      CTOC  = CENTRE-TO-CENTRE DISTANCE BETWEEN TOP & BOTTOM ROLLS
C
C      N.B.:- GIVE ONLY ONE VALUE EACH OF IPASHT=+1  ,OTHERWISE
C      GIVE THE CORRECT NUMBER (= NSTAGE) OF VALUES EACH IF IPASHT=-1;
C      BOTH PASHT & CTOC VALUES MUST BE POSITIVE.
C
C  R3. ITOL =+1 (IF CONSISTENT LH & RH TOLERANCES),OR
C           -1 (IF INCONSISTENT LH & RH TOLERANCES).
C
C  R4. TOLLH = LEFT-HAND TOLERANCE BETWEEN ROLLER AND LAST ELEMENT
C      TOLLR = RIGHT-HAND TOLERANCE BETWEEN ROLLER AND LAST ELEMENT.
C
C      N.B.:- GIVE ONLY ONE VALUE EACH IF ITOL=+1, OTHERWISE
C             GIVE THE CORRECT NUMBER (= NSTAGE) OF VALUES EACH
C             IF ITOL=-1; BOTH TOLLH AND TOLRH MUST BE POSITIVE.
C
C  R5. JTEMP = 1 IF COMPONENT DRAWING IN HIDDEN-LINE FORM
C                IS REQUIRED, OTHERWISE 0
C
C      RSCALE = DESIRED SCALE FOR THE ROLLER DRAWINGS (AUTOMATICALLY
C               REDUCED IF IT EXCEEDS THE MAXIMUM PERMISSIBLE SCALE)
C
C      RGAP = THE GAP DIMENSION BETWEEN TOP AND BOTTOM ROLLERS
C
C
C  R6. JSTAGE(1)=-1 (IF ALL STAGES REQUIRED FOR ROLLERS),OR
C      JSTAGE(1)= 0 (IF NO STAGES REQUIRED FOR ROLLERS),OR
C      JSTAGE(1)=ANY POSITIVE VALUE(INTEGER LESS THAN NSTAGE) (IF
C               SELECTED STAGES REQUIRED FOR ROLLERS).
C
C      N.B.:- IF JSTAGE(1) IS -1 OR 0, NO FURTHER INPUT DATA IS REQUIRED;
C             IF JSTAGE(1) IS POSITIVE,THEN SUPPLY OTHER DESIRED
C             BENDING STAGE NO. IN ASCENDING ORDER, ENTER 0 TO TERMINATE.
C             THE TOTAL NUMBER OF JSTAGE DATA MUST NOT EXCEED
C             NSTAGE VALUE.
C
```

```
C*************************************************************************
C
C      SPECIAL OPTIONS
C
C*************************************************************************
C
C      PINCH-DIFFERENCE OR DRIVE/CLEARANCE SURFACE OPTION:-
C      ====================================================
C
C P1. JPINCH = 0 (IF NO EXTERNAL PINCH DIFFERENCE DIMENSION
C                 DEFINITION IS SUPPLIED), OR
C              1 (IF ONLY ONE PINCH DIFFERENCE DIMENSION DEFINITION
C                 IS SUPPLIED FOR ALL BENDING STAGES), OR
C             -1 (IF PINCH DIFFERENCE DIMENSION DEFINITION IS SUPPLIED
C                 INDIVIDUALLY FOR EACH BENDING STAGE).
C
C              2 (IF SINGLE THICKNESS OPTION WITH ONE COMMON CLEARANCE
C                 VALUE FOR ALL STAGES SELECTED),OR
C
C             -2 (IF SINGLE THICKNESS OPTION WITH INDIVIDUAL CLEARANCE
C                 VALUE FOR EACH STAGE SELECTED)
C
C      N.B. :- IF JPINCH IS 0,1 OR -1 THEN PROCEED TO SECTION P2.
C              IF JPINCH IS 2 OR -2 THEN PROCEED TO SECTION P2A.
C
C
C P2. PDIM1 = THICKNESS BETWEEN THE DRIVE SURFACES.
C     PDIM2 = THICKNESS BETWEEN THE CLEARANCE SURFACES.
C
C      N.B. :- IF JPINCH IS 0, DATA FOR PDIM1 AND PDIM2 ARE NOT REQUIRED;
C              IF JPINCH IS 1, ONLY 1 SET IS REQUIRED; AND
C              IF JPINCH IS -1, 1 SET FOR EACH STAGE SHOULD BE SUPPLIED.
C
C P3. ISQP = CURRENT BENDING STAGE SEQUENCE NO.
C     IEPL = L.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR.
C     IEPR = R.H.S. ELEMENT DEFINING THE DRIVE SURFACE CONTOUR.
C
C      N.B. :- TO TERMINATE DEFINITION STATEMENT SEQUENCE, ENTER 0 FOR ISQP,
C              AND DUMMY VALUES FOR THE OTHER ITEMS.
C              TWO STATEMENTS REQUIRED FOR EACH BENDING STAGE
C              TO DEFINE THE BEGINNING AND THE END OF DRIVE-SURFACE.
C              TOTAL NO. OF DEFINITION STATEMENT SHOULD BE 50 OR LESS.
C              IF ONLY EITHER L.H.S. OR R.H.S. IS TO BE DEFINED THEN THE
C              NON-APPLICABLE SIDE SHOULD HAVE ELEMENT 0 ENTERED.
C
C P2A. **  THIS IS THE SINGLE THICKNESS OPTION FOR H&E **
C       CLEAR = CLEARANCE VALUE TO BE ADDED ONTO THE THICKNESS WITH
C               8 PER CENT OF THICKNESS ALREADY ADDED .
C
C       N.B. :- IF JPINCH IS 2, THEN SUPPLY ONLY ONE VALUE;
C               IF JPINCH IS -2, THEN SUPPLY THE CORRECT NUMBER (=NSTAGE)
C               OF VALUES FOR EACH INDIVIDUAL STAGE.
C
C-------------------------------------------------------------------
C
C      SIDE-CONTOUR OPTION:-
C      =====================
C
C S1. ISROL = 1 (IF SIDE-ROLL OPTION WILL BE USED), OR
C             0 (IF SIDE-ROLL OPTION WILL NOT BE USED)
C
C      N.B. IF ISROL IS 0, THEN NO FURTHER DATA IS REQUIRED FOR
C           OTHER SIDE-ROLL INPUT AND EXTENSION-CONTOUR INPUT.
C
C S2. IOTOS = 1 (IF CONSISTENT ORIGIN TO SIDE-ROLL AXES) OR
C             -1 (IF INCONSISTENT ORIGIN TO SIDE-ROLL AXES DISTANCES)
C
C S3. OTOSL = DISTANCE BETWEEN ORIGIN AND LEFT SIDE-ROLL CENTRE
C     OTOSR = DISTANCE BETWEEN ORIGIN AND RIGHT SIDE-ROLL CENTRE
C
C      (SEE DIAGRAM ).
C
C      N.B. :- GIVE ONLY 1 SET OF VALUES IF IOTOS IS 1; IF
C              IOTOS IS -1 THEN GIVE CORRECT NUMBER (=NSTAGE) OF
C              SETS OF VALUES FOR EVERY STAGE; BOTH OTOSL AND OTOSR
C              MUST BE POSITIVE.
C
```

```
C S4. ISQS  = CURRENT BENDING STAGE SEQUENCE NO.
C     IESL1 = L.H.S. ELEMENT CONSTITUTING L.H.S. SIDE-ROLL CONTOUR.
C     IESL2 = WHICH FACE OF THE L.H.S. ELEMENT IS DESIRED
C     IESR1 = R.H.S. ELEMENT CONSTITUTING R.H.S. SIDE-ROLL CONTOUR.
C     IESR2 = WHICH FACE OF THE R.H.S. ELEMENT IS DESIRED
C
C     (SEE DIAGRAM FOR DETAIL OF ELEMENT FACE DEFINITION ).
C
C     N.B. :- IF SIDE-ROLLS ARE NOT REQUIRED FOR A PARTICULAR
C             BENDING STAGE, THEN SKIP AND GO ON TO DEFINE THE
C             NEXT STAGE WHICH HAS SIDE-ROLLS.
C             THE SIDE-ROLL CONTOUR IS DEFINED BY 2 BOUNDARY
C             ELEMENT FACES, NAMELY THE STARTING ELEMENT FACE AND
C             THE ENDING ELEMENT FACE. THEY MUST BE ENTERED IN THAT
C             SEQUENCE IN 2 CONSECUTIVE DEFINITION STATEMENTS.
C             THE DEFINITION CONVENTION IS FROM BOTTOM UPWARDS, HENCE
C             THE COMPLETE SIDE-ROLL CONTOUR WILL CONSIST OF THE BOTTOM
C             LEADING PART, FOLLOWED BY THE PART IN CONTACT WITH THE
C             STRIP AND THEN FOLLOWED BY THE TRAILING TOP PART.
C             NO MORE THAN 2 DEFINITION STATEMENTS SHOULD BE ENTERED
C             FOR EACH SIDE-ROLL.
C             TO TERMINATE DEFINITION STATEMENT SEQUENCE,
C             ENTER 0 FOR ISQS,AND DUMMY VALUES FOR THE OTHER ITEMS.
C             IF ONLY EITHER L.H.S. OR R.H.S. ELEMENT IS TO BE DEFINED
C             IN THE CURRENT STATEMENT, THEN ENTER 0 FOR THE ELEMENT
C             ON THE NON-APPLICABLE SIDE.
C             TOTAL NO. OF DEFINITION STATEMENTS SHOULD BE 50 OR LESS.
C
C
C
C S5. ** THIS SECTION ON EXTENSION-CONTOUR OPTION CAN BE USED
C       ONLY IF SIDE-ROLL OPTION IS USED, ELSE IT IS IRRELEVANT **
C
C        IEXTN = 1 (IF EXTENSION-CONTOUR OPTION WILL BE USED), OR
C                0 (IF EXTENSION-CONTOUR OPTION WILL NOT BE USED).
C
C        N.B. :- IF IEXTN IS 0, THEN NO FURTHER DATA INPUT FOR THIS
C                OPTION IS REQUIRED.
C
C S6. ISC1 = SIDE-CONTOUR NO.
C     ISC2 = SIDE-CONTOUR TYPE (1 FOR SIDE-ROLL, 2 FOR SPIGOT TYPE A,
C                               3 FOR SPIGOT TYPE B.)
C     SCDIM1 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 1
C     SCDIM2 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 2
C     SCDIM3 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 3
C     SCDIM4 = DIMENSION DEFINITION FOR SIDE-CONTOUR PART 4
C
C     (SEE DIAGRAM FOR DETAILS OF EACH DEFINITON ).
C
C S7. ISQD  = BENDING STAGE NO. FOR WHICH SIDE-CONTOUR DEFINITION
C             IS INTENDED
C     IELD1 = TYPE OF SIDE-CONTOUR ON L.H.S.
C     IELD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
C             IS TO BE SELECTED FOR THIS STAGE NO. ON L.H.S.
C     IERD1 = TYPE OF SIDE-CONTOUR ON R.H.S.
C     IERD2 = SIDE-CONTOUR NO. WHICH HAS BEEN DEFINED PREVIOUSLY AND
C             IS TO BE SELECTED FOR THIS STAGE NO. ON R.H.S.
C     N.B. :- THE SIDE-CONTOUR TYPE MUST MATCH WITH THE DEFINED TYPE
C             DESCRIBED IN SECTION S2.
C             IF THE SELECTED CONTOUR TYPE IS 1 (I.E. SIDE-ROLL),
C             THEN 2 DEFINITION STATEMENTS FOR BOTTOM LEADING PART AND
C             TOP TRAILING PART RESPECTIVELY MUST BE ENTERED IN
C             SUCCESSION WHEREAS IF THE SELECTED CONTOUR TYPE IS 2 OR 3
C             (I.E. SPIGOTS),THEN ONLY 1 DEFINITION STATEMENT IS REQUIRED.
C
```

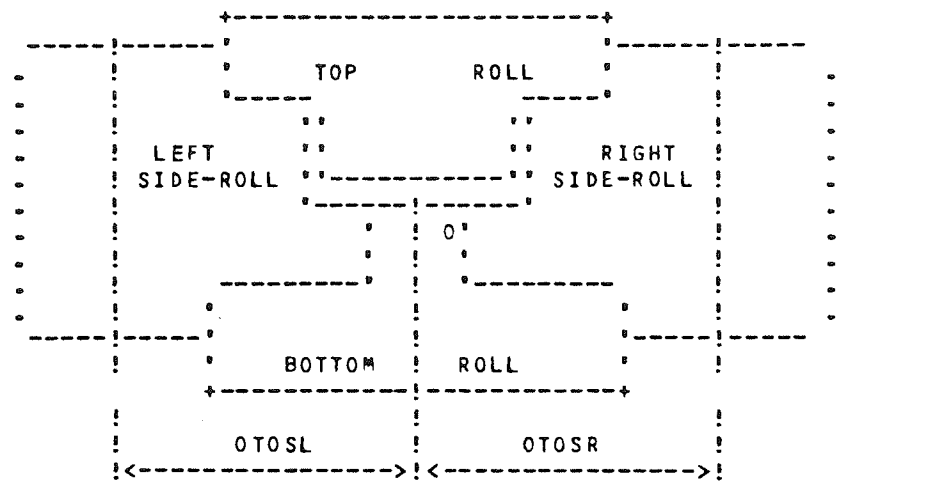
#LISTING OF :EPP3136.DIAGRAM(1/) PRODUCED ON 6NOV81 AT 20.45.56

#G8.64AB AT ASTON IN :EPP3136.MOP1° ON 11NOV81 AT 14.51.38 USING U15

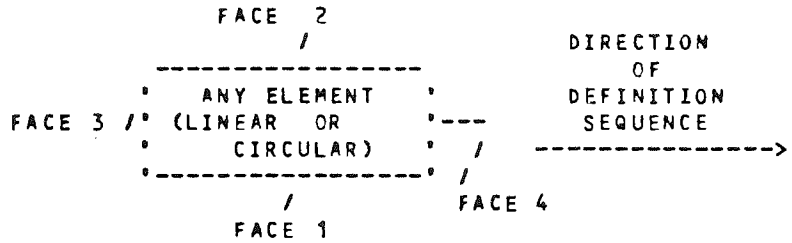
DOCUMENT DIAGRAM

DIAGRAMS FOR ROLL INPUT DATA FILE NOTES IN R-INPUT-INFO
****

(1) DISTANCES BETWEEN ORIGIN AND SIDE-ROLL AXES
==========



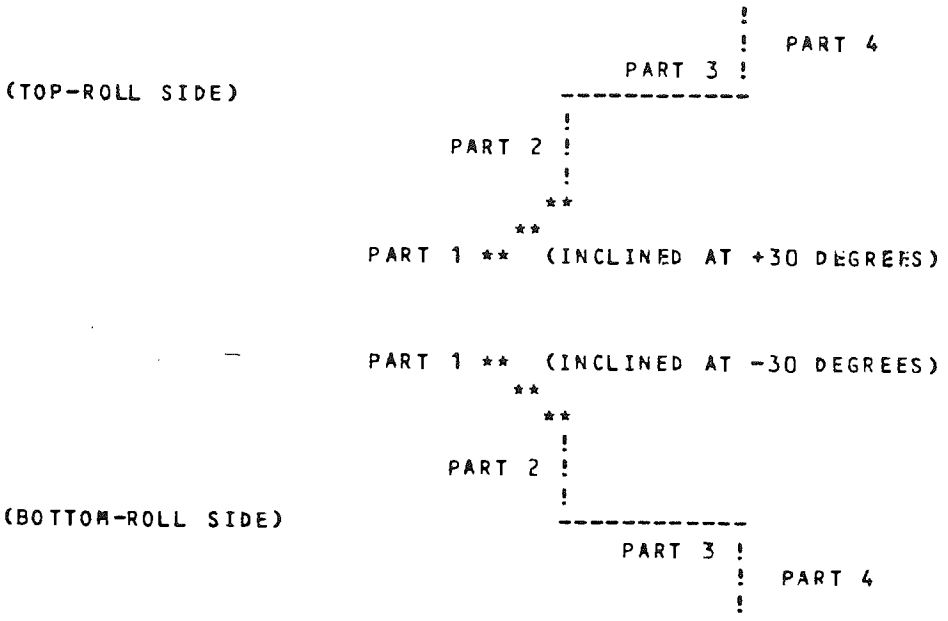
(2) ELEMENT FACES FOR DEFINING SIDE-ROLL CONTOURS
==========



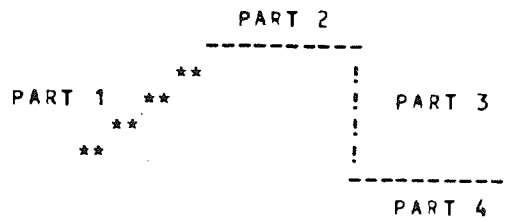
WHERE FACE 1 IS BOTTOM FACE,
FACE 2 IS TOP FACE,
FACE 3 IS LEADING FACE AND
FACE 4 IS TRAILING FACE.

(3) PARTS FOR DEFINING EXTENSION-CONTOURS

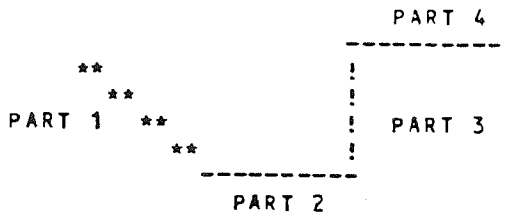
TYPE 1 :- SIDE-ROLL EXTENSION-CONTOURS



TYPE 2 :- SPIGOT TYPE A



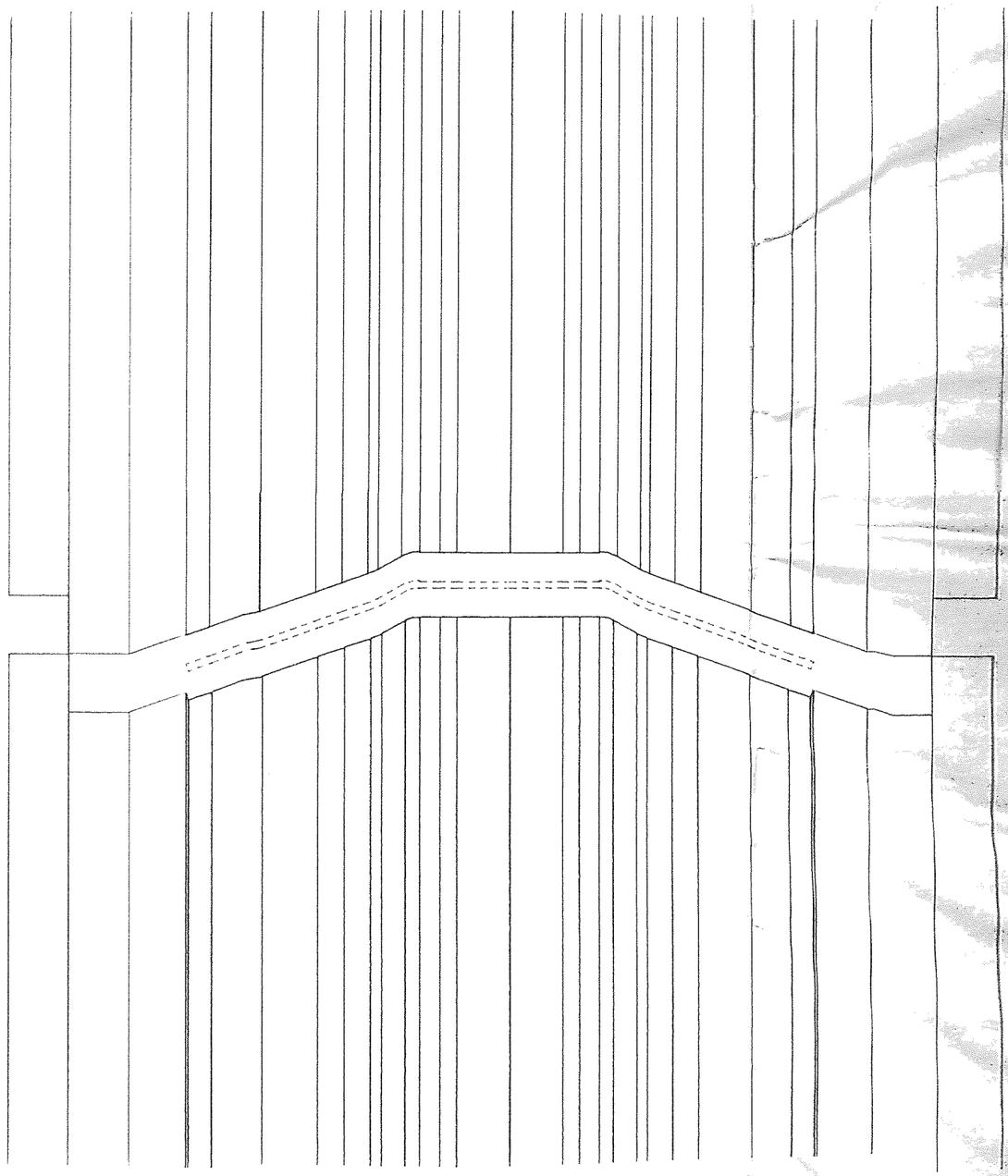
TYPE 3 :- SPIGOT TYPE B



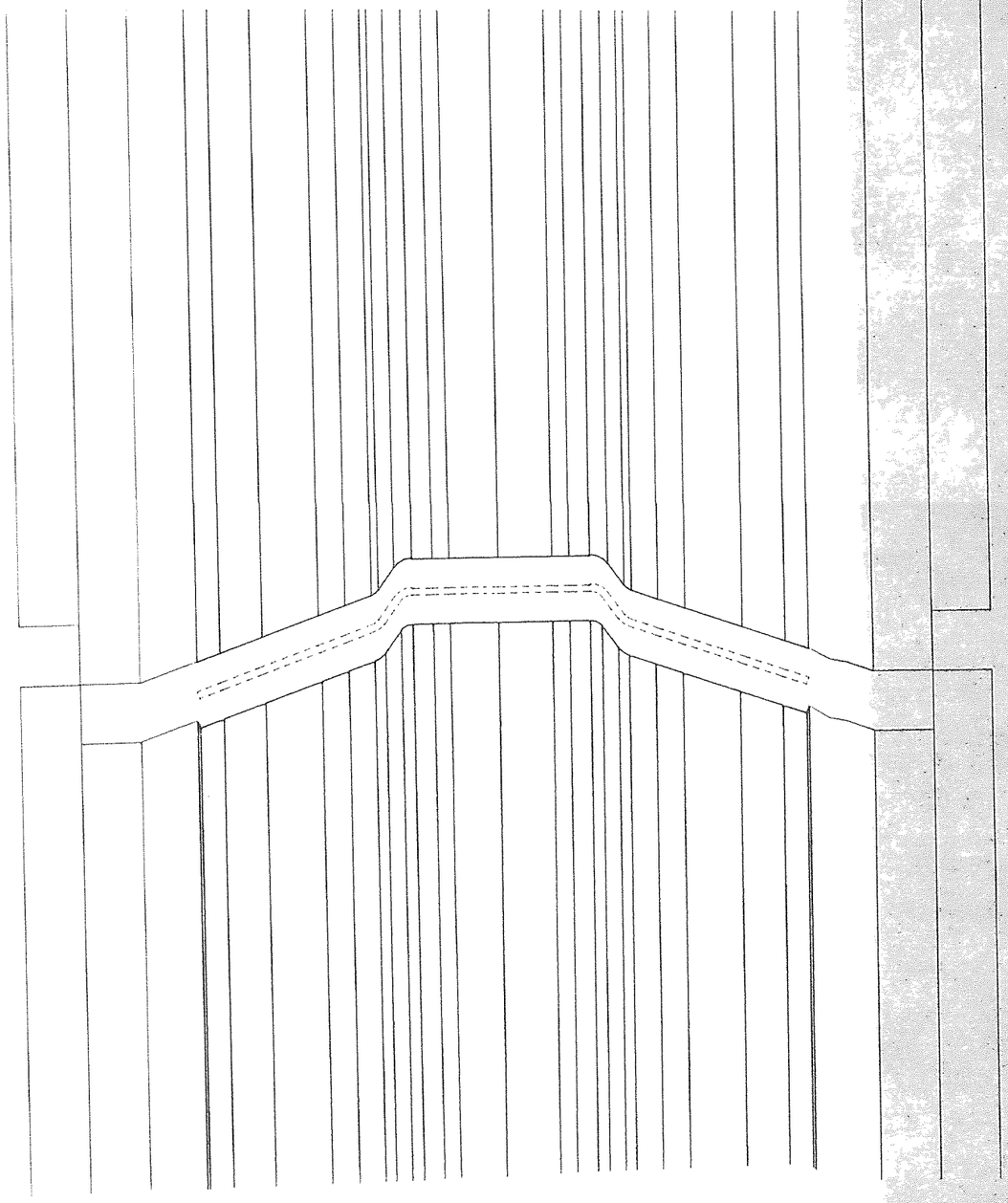
(N.B. :- THE IRRELEVANT PARTS OF THE EXTENSION-CONTOURS MAY HAVE THEIR LENGTH MADE ZERO)

APPENDIX 4

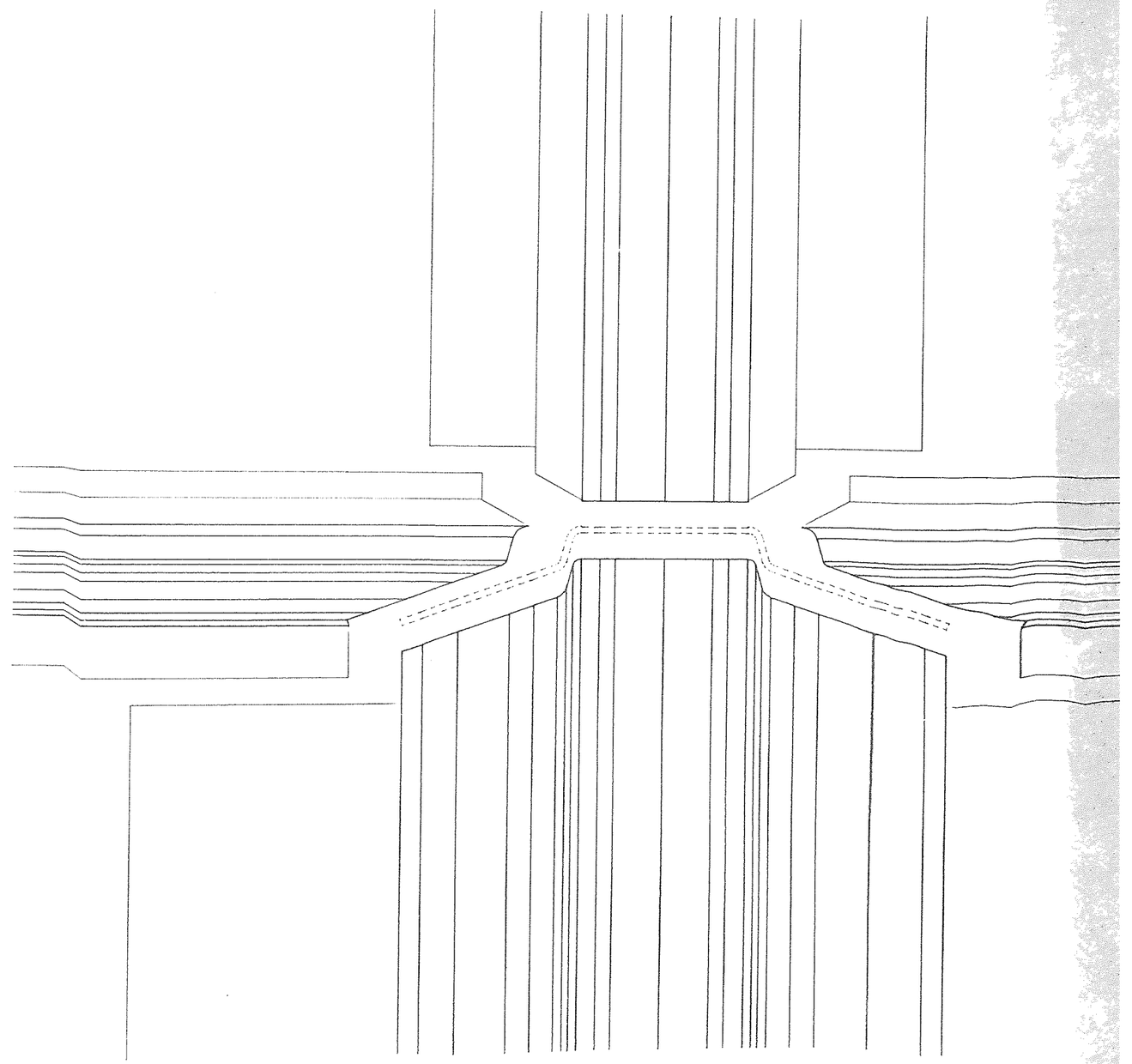
A PRACTICAL EXAMPLE OF ROLL DESIGN DRAWINGS GENERATED DURING THE JOB RUN



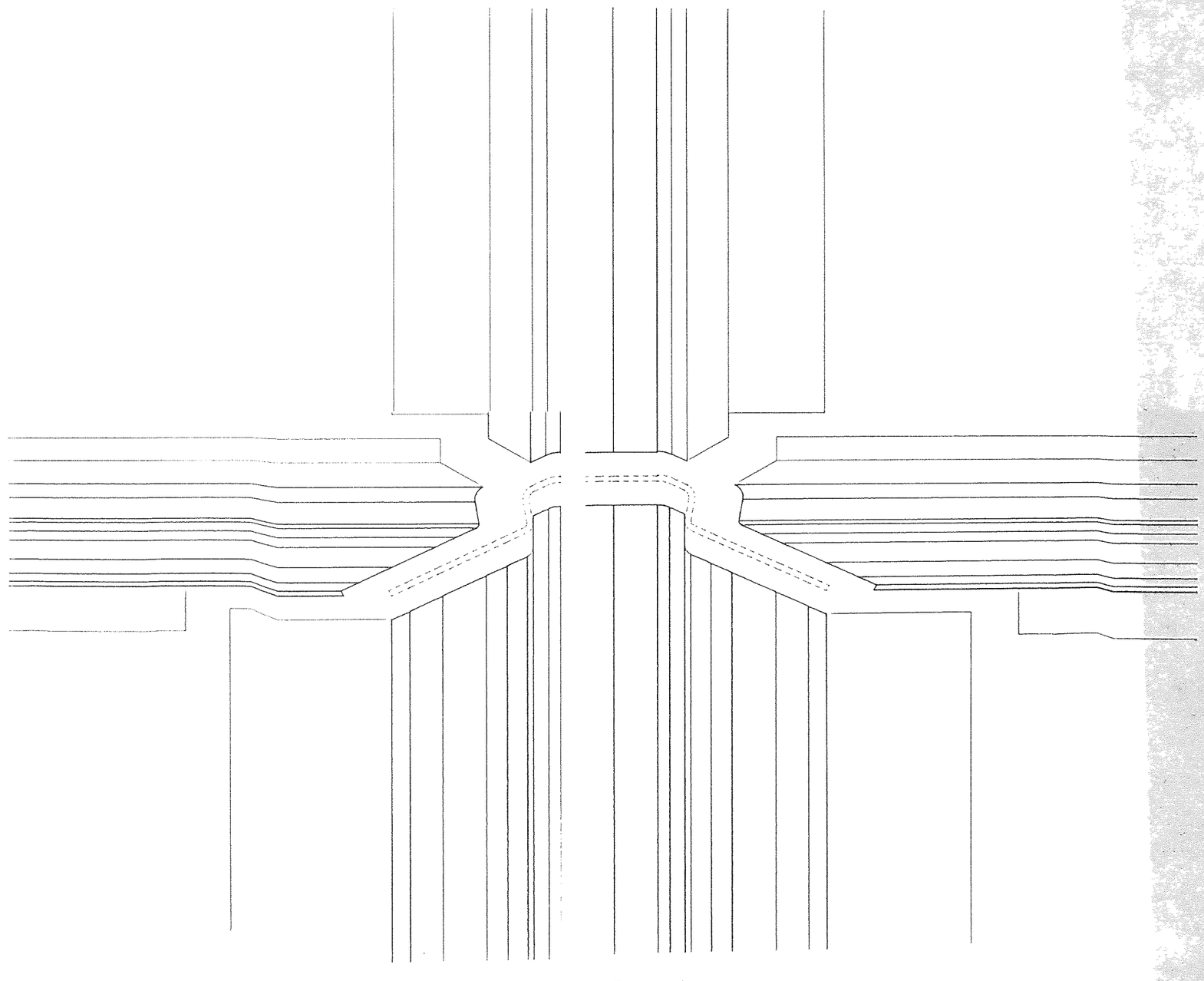
STAGE 1



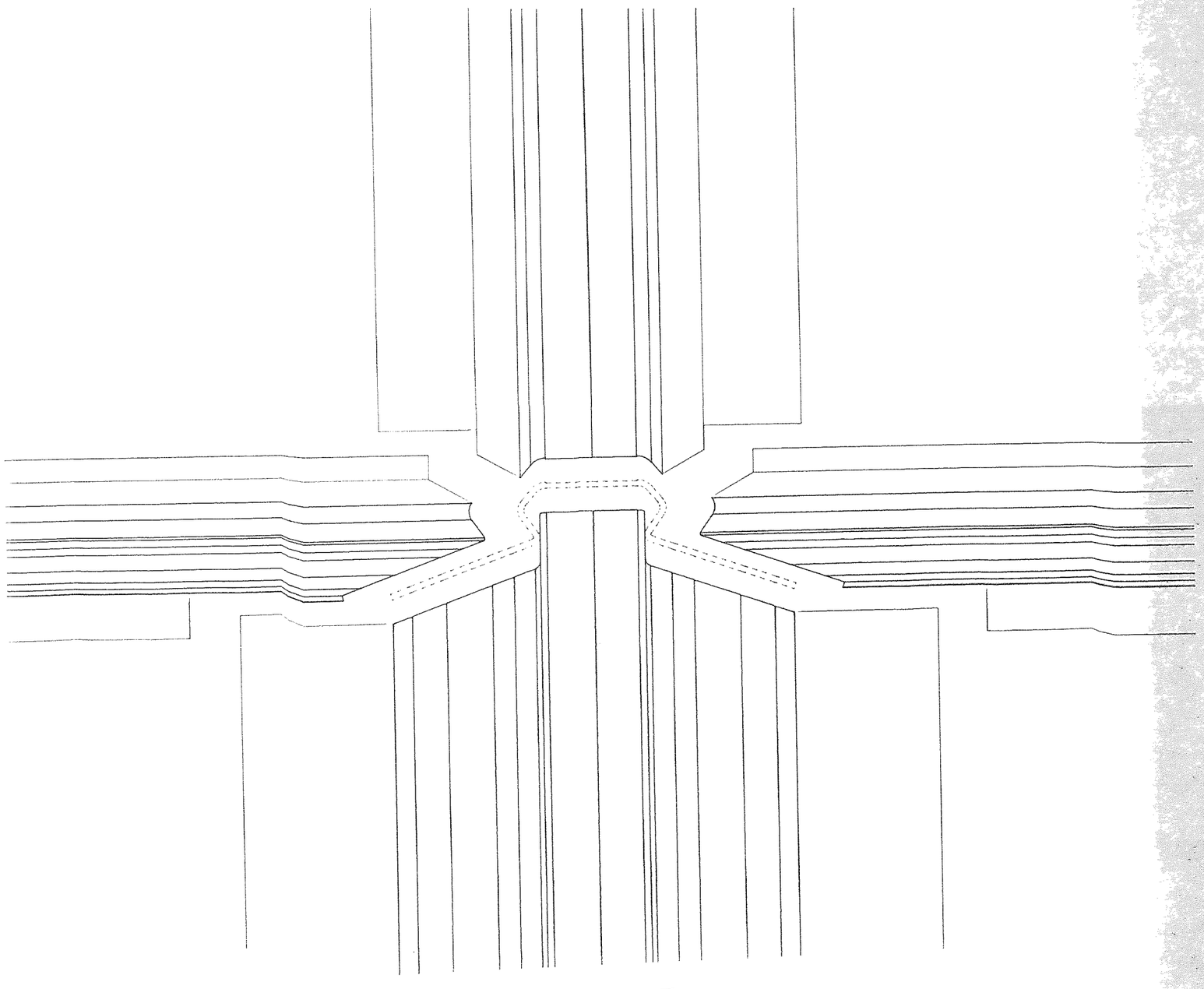
STAGE 2



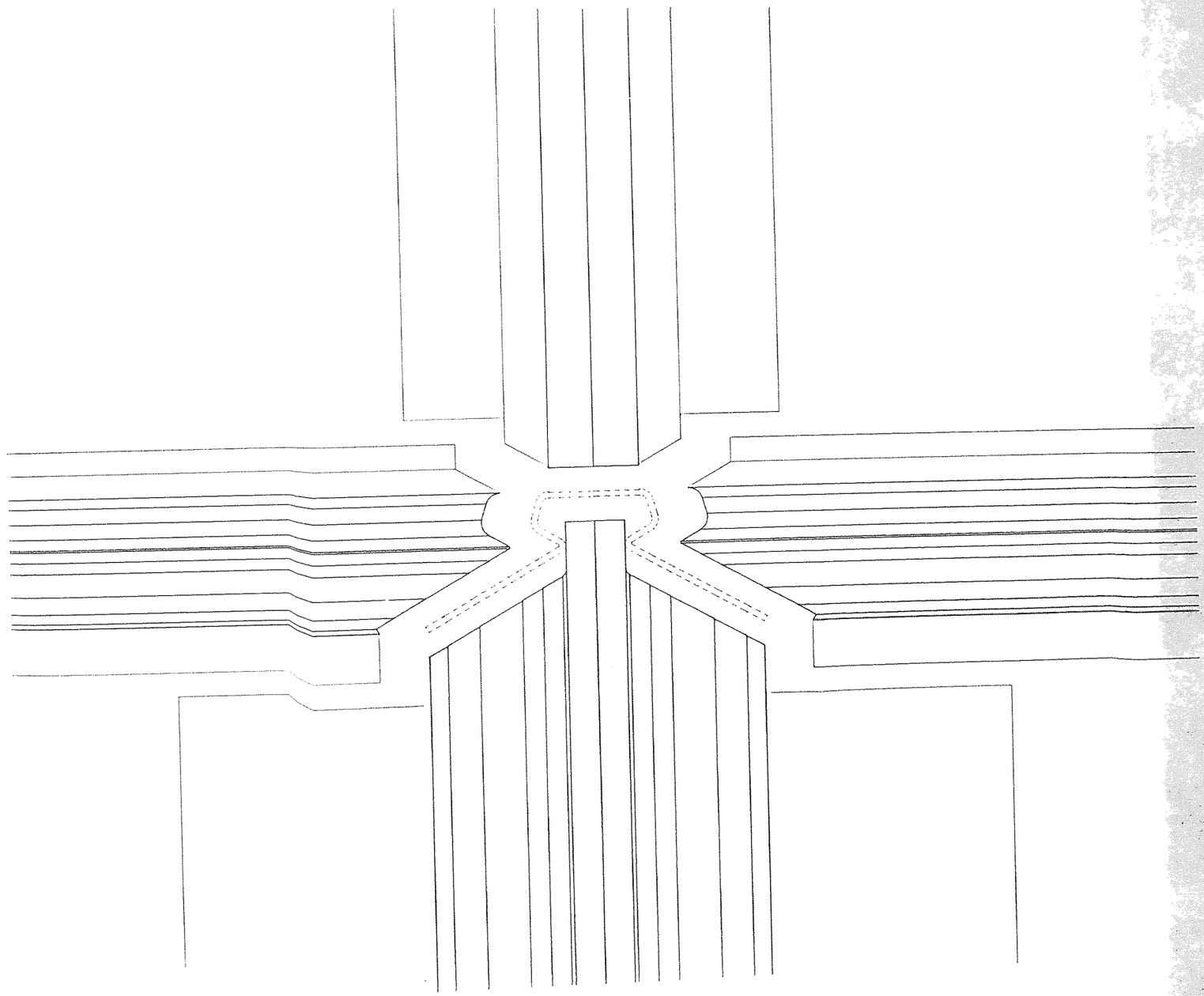
STAGE 3



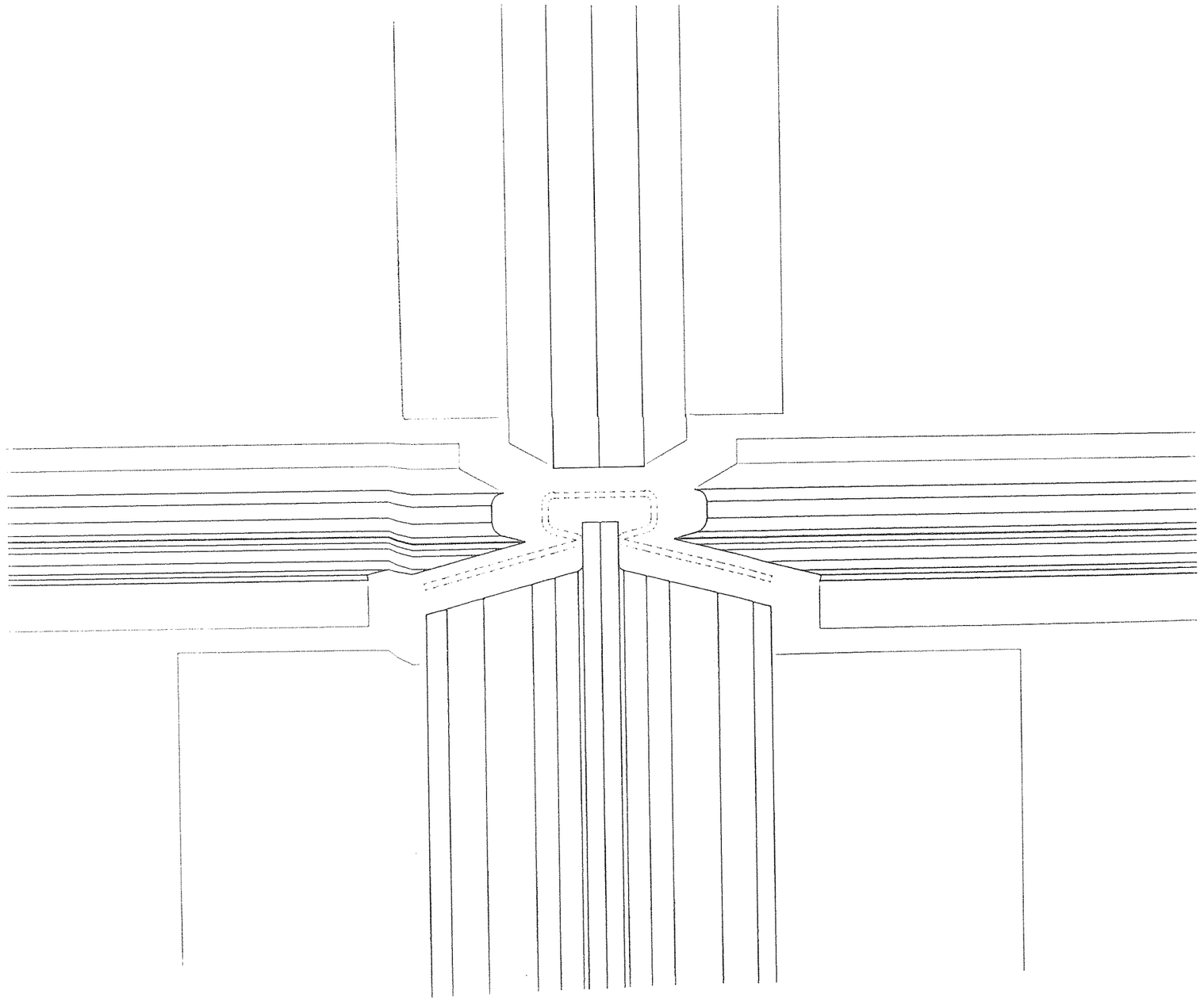
ST GE 4



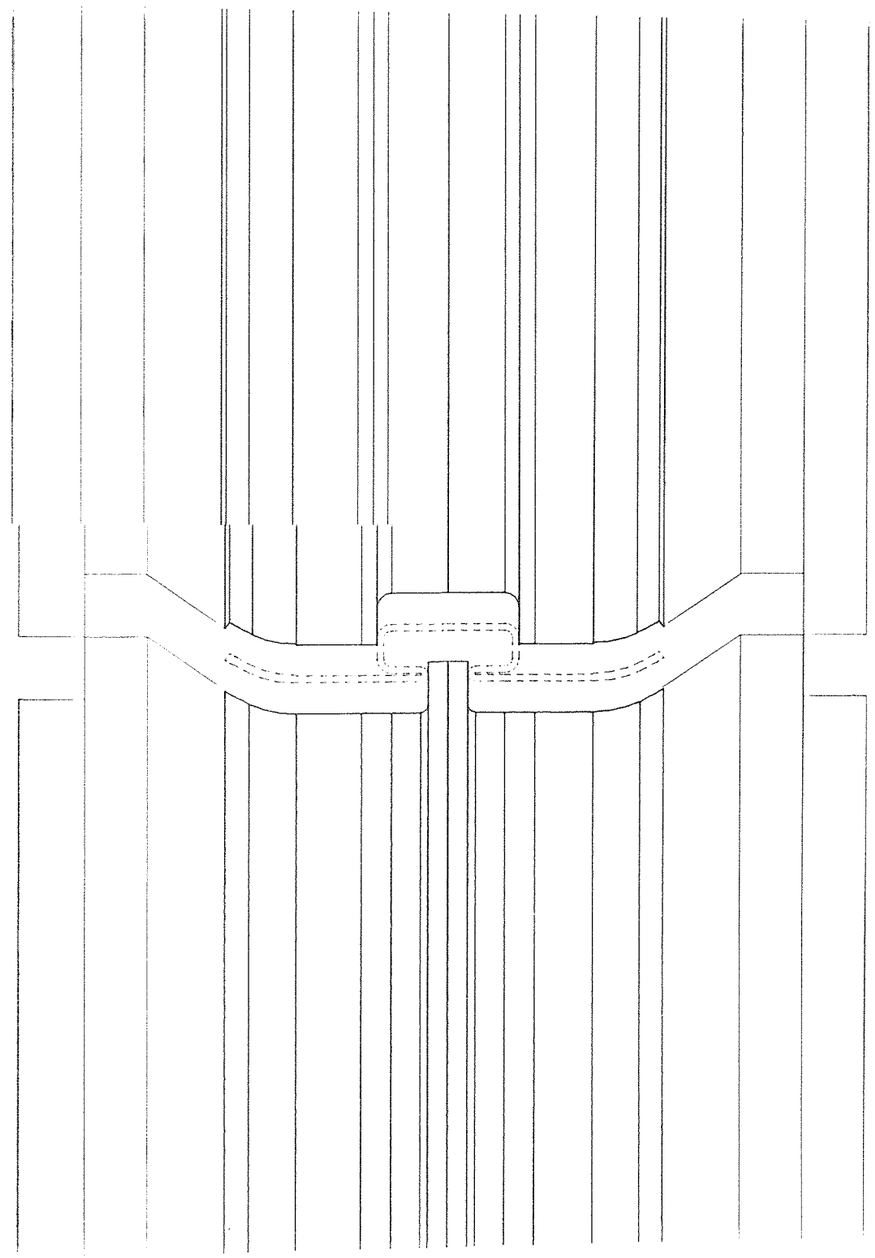
STAGE 5



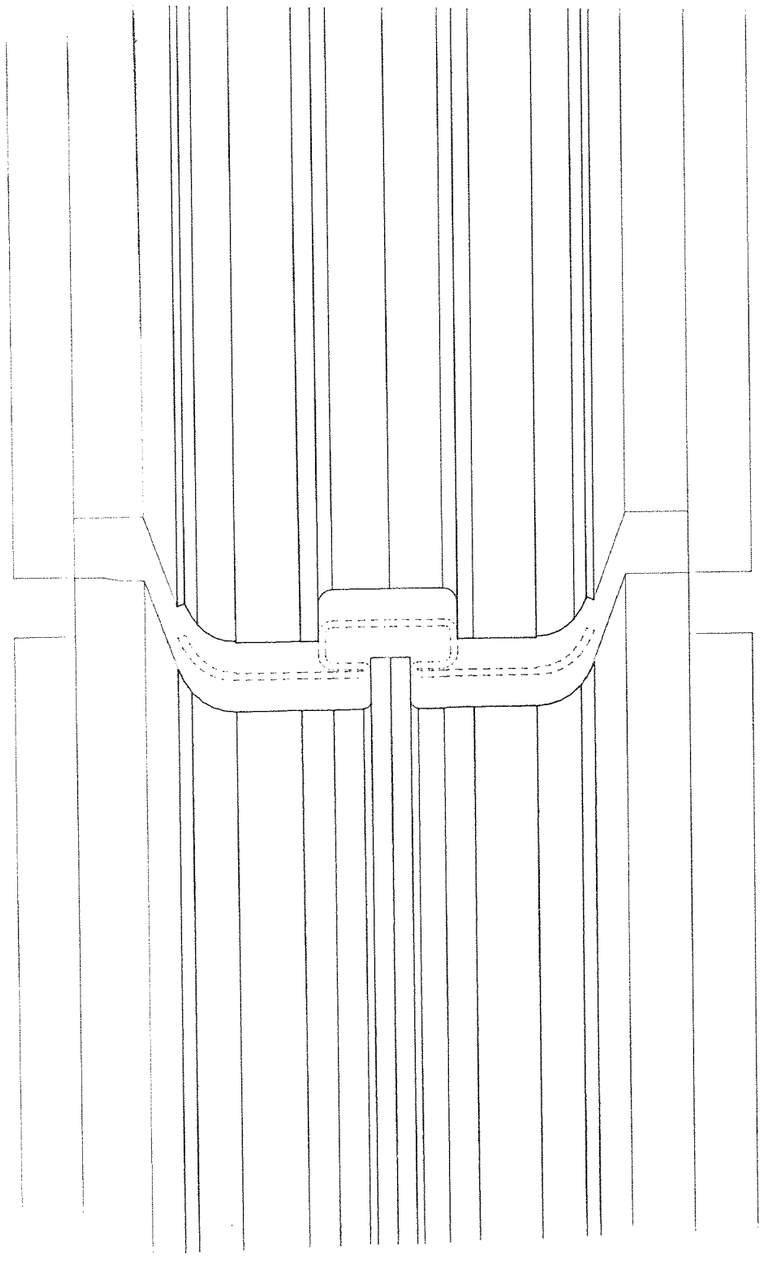
STAGE 6



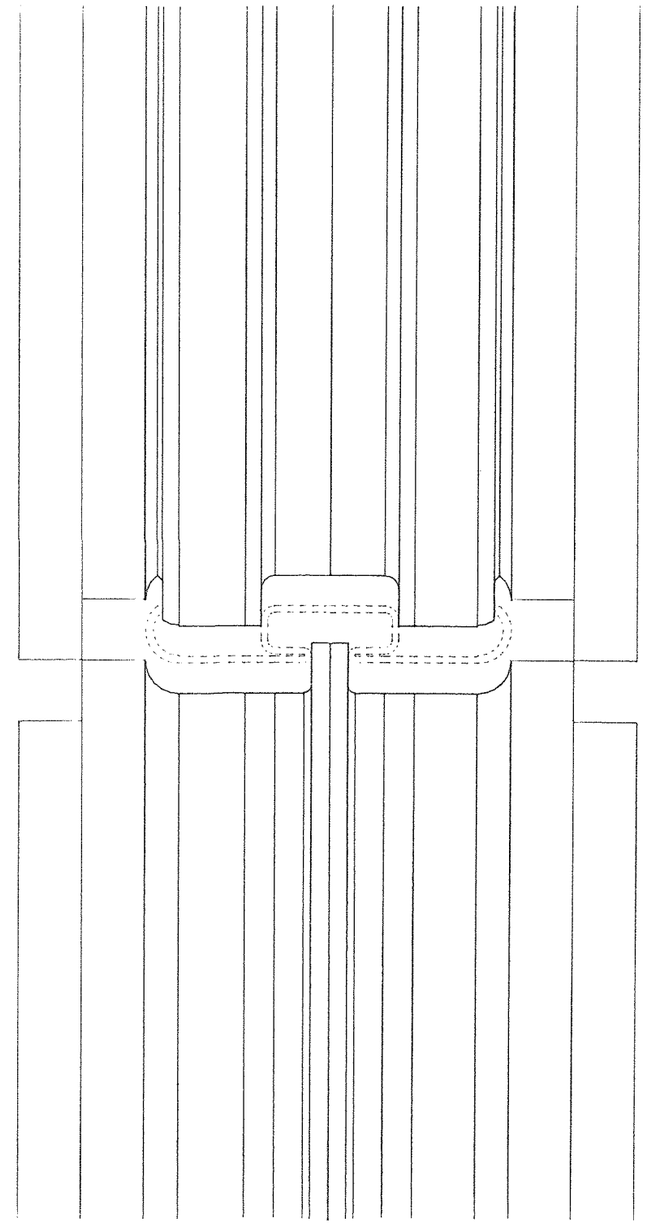
STAGE 7



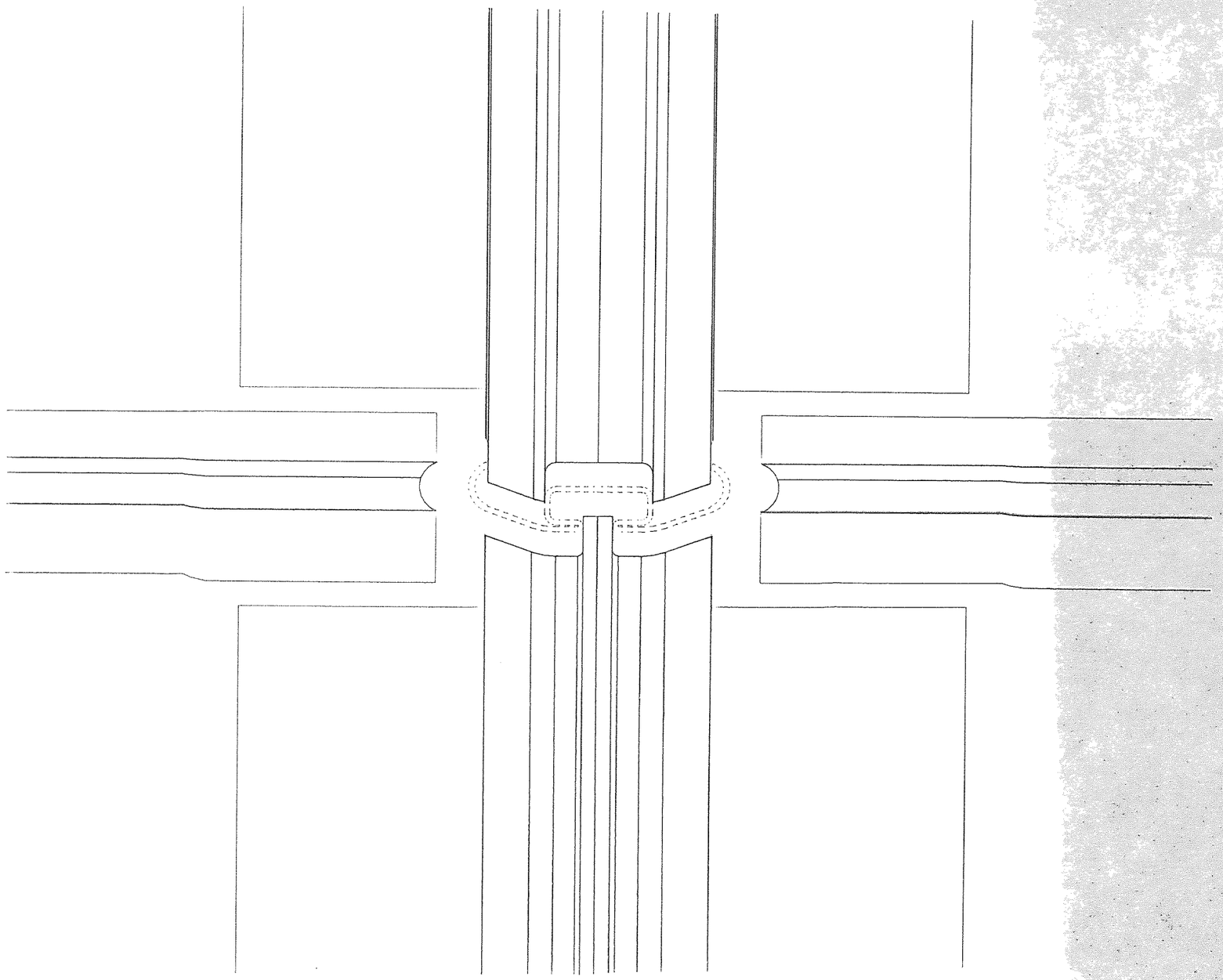
STAGE 8



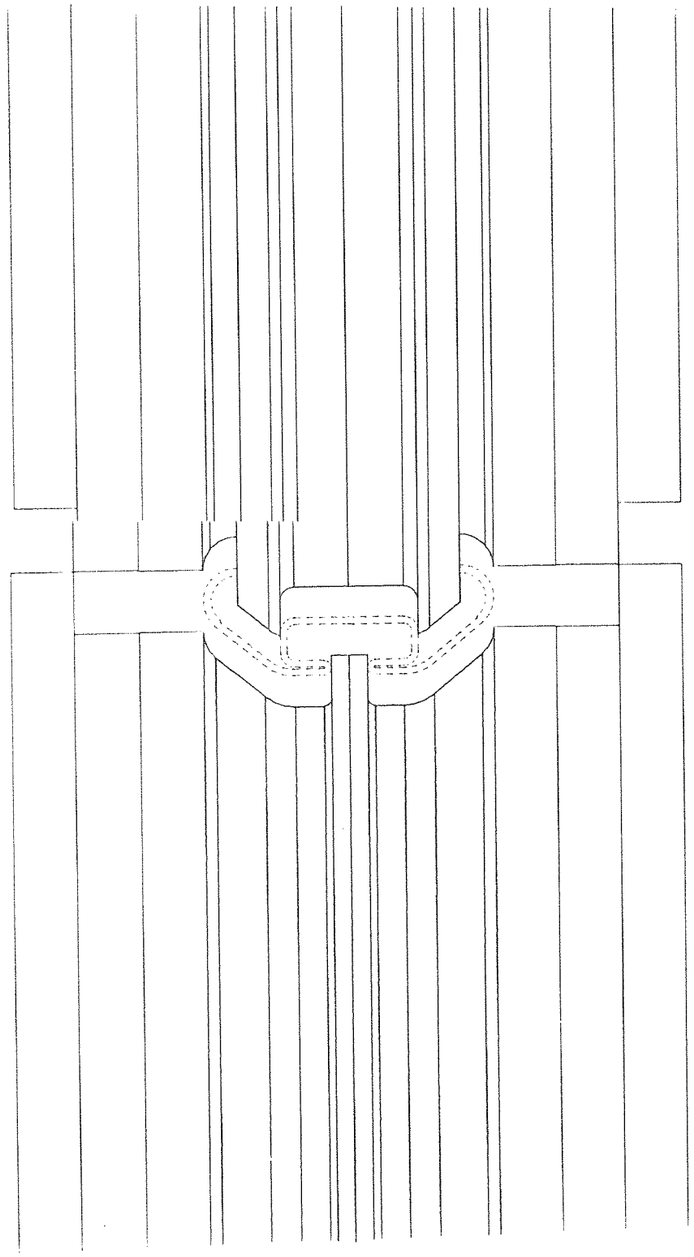
STAGE 9



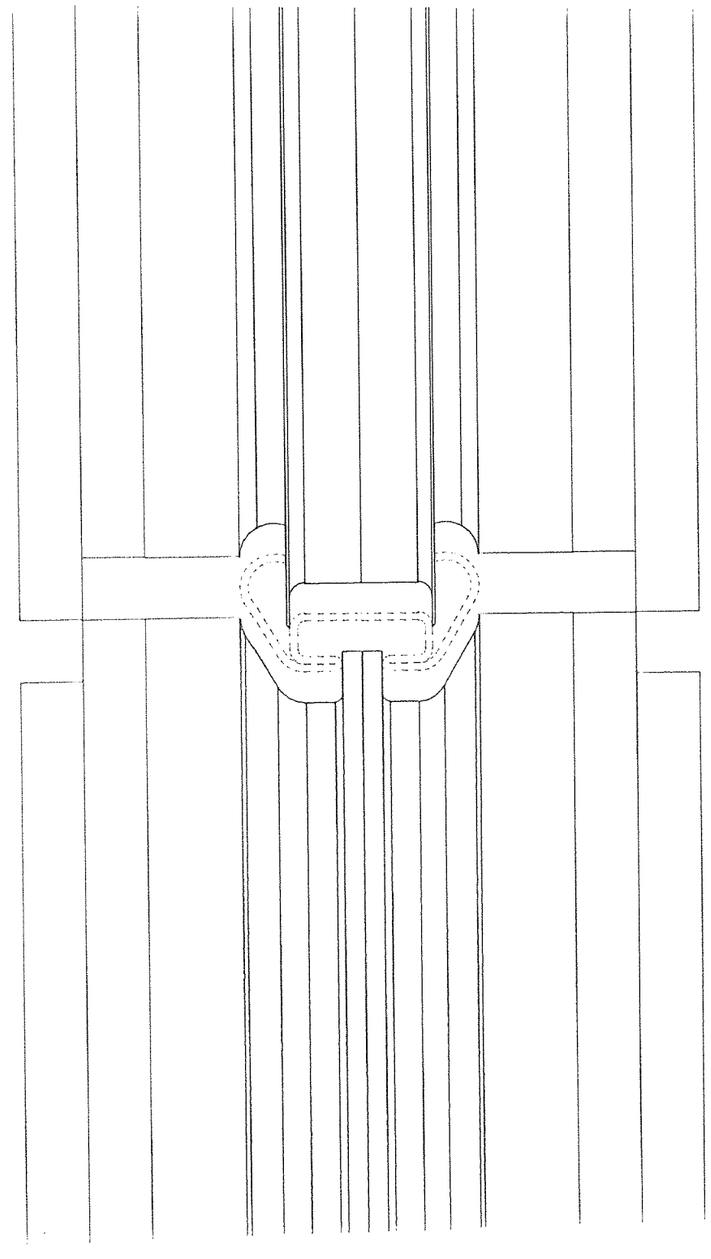
STAGE 10



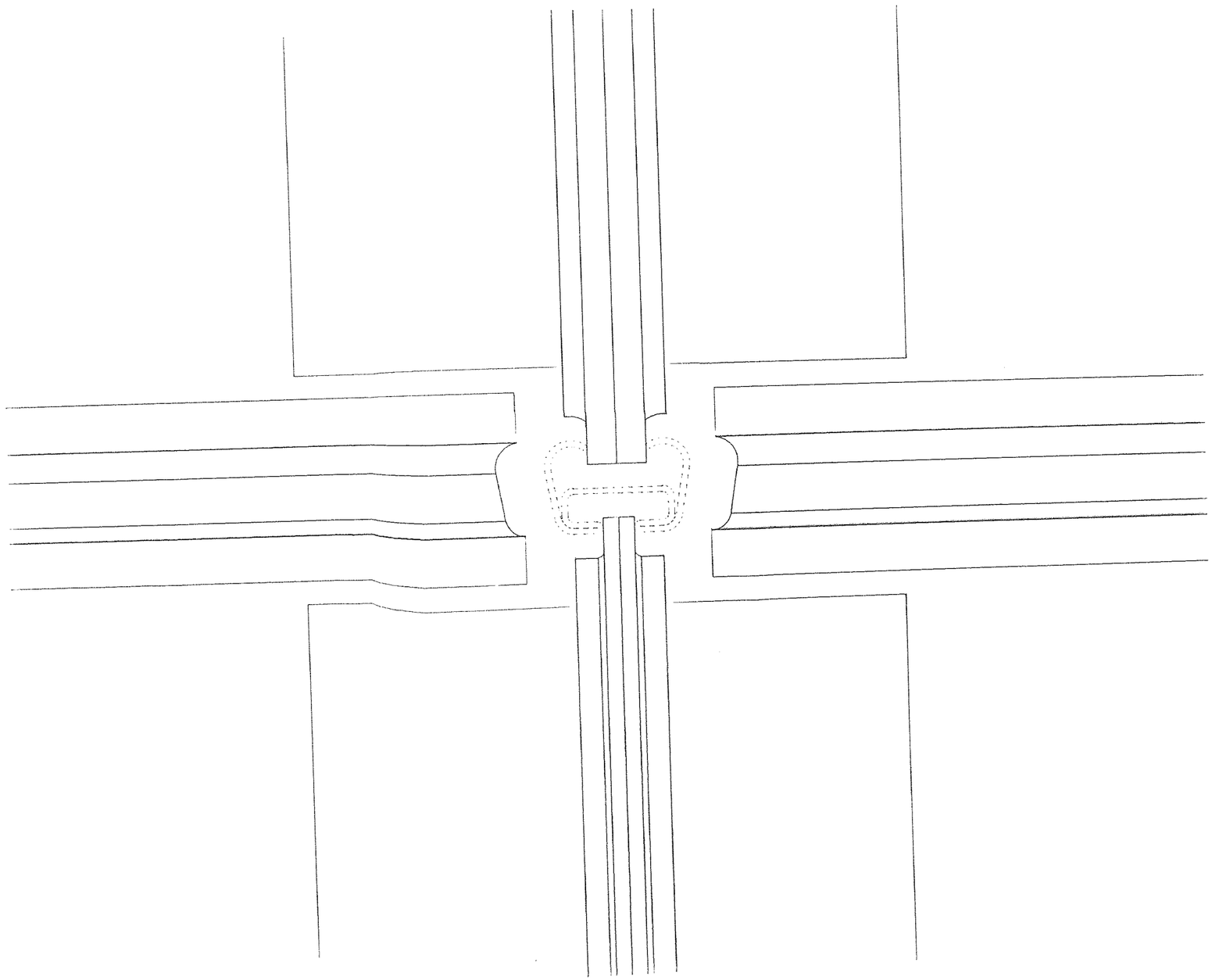
STAGE 11



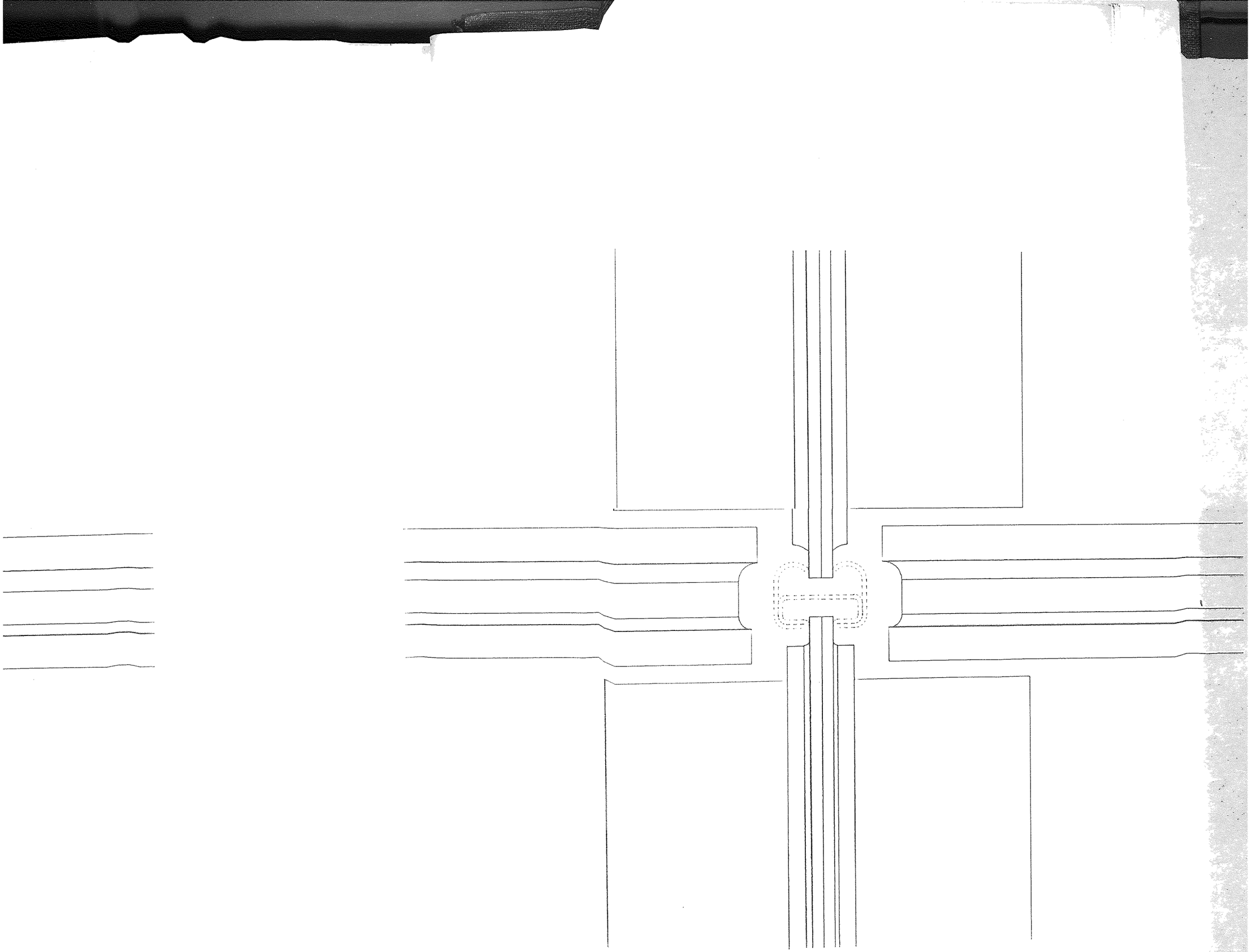
STAGE 12



STAGE 13



STAGE 14



STAGE 15

APPENDIX 5

PROGRAM LISTINGS*

1. THE FINISHED SECTION PROGRAM (RPLOT1)

2. THE FLOWER PATTERN PROGRAM (RPLOT2)

3. THE 10 TO 1 TEMPLATE PROGRAM (RPLOT3)

4. THE ROLL DESIGN PROGRAM (RPLOT4)

* Refer to the microfiche attached at the back cover.

REFERENCES

1. Coons, S.A. "Surfaces for Computer Aided Design", Design Division, Mech. Eng. Dept., M.I.T., Cambridge, Massachusetts, 1964.

2. Forrest, A.R. "Curves and Surfaces for Computer Aided Design", Ph.D. Thesis, University of Cambridge, 1968.

3. Bezier, P.E. "How Renault Uses NC for Car Body Design and Tooling", (SAE Paper 680010) Proc. Soc. of Automotive Engineers Congress, Detroit, 1968.

4. Leslie, W.H.P. "CAD/CAM in U.K. Industry", Computer Aided Design, Vol. 13 No. 3, May 1981, pp 166 to 170.

5. "Cold Roll-Formed Sections", Cold Rolled Sections Association, U.K., 1980.

6. American Society of Manufacturing Engineers, "Tool and Manufacturing Engineers' Handbook - Cold Roll Forming", McGraw Hill Book Co., 1976, pp 15-139 to 15-153.

7. American Society of Mechanical Engineers, "ASME Handbook - Metals Engineering Process (Cold Roll Forming)", McGraw Hill Book Co., 1958, pp 167 to 177.

8. American Society for Metals, "Metals Handbook - Vol. 4 (Forming)", 1969, pp 224 to 239.

9. BS 1449 "Part 1 Carbon steel plate, sheet and strip".

10. BS 4360 "Specification for weldable structural steels".

11. BS 1449 "Part 2 Stainless steel and heat-resisting steel plate, sheet and strip".

12. BS 2989 "Hot-dip zinc coated steel sheet and coil".

13. BS 2920 "Cold reduced tin plate and cold reduced black plate".

14. BS 1470 "Wrought alluminium and alluminium alloys for general engineering purposes - plate, sheet and strip".

15. BS 2870 "Rolled copper and copper alloys - sheet, strip and coil".

16. BS 4608 "Copper for electrical purposes - rolled sheet, strip and foil".

17. Walker, A.C. (Editor) "Design and Analysis of Cold Formed Sections", Intertext Books, 1975.

18. "Thin Walled Steel Structures", Proceedings of the 1967 Symposium at University College of Swansea, Crosby and Lockwood.

19. Rhode, J. and Walker, A.C. (Editors) "Thin Walled Structures", Proceedings of the 1979 International Conference on Cold Formed Sections at University of Strathclyde, Glasgow.

20. Bryan, E.R. "The Stressed Skin Design of Steel Buildings", Crosby and Lockwood, 1972.

21. Kaltprofile, Verlag Stahleisen MBH, Dusseldorf, 1969.

22. "International Directory of Software 1980-1981", Computer Users' Year Book Publications Ltd., 1980.

23. Rhodes, A. "Computer-Aided Roll Design for Cold Roll Forming", Production Engineer, September 1981, pp 32 to 34.

24. Teicholz, E. "Mini-Based Turnkey Graphic Systems", Datamation, March 1980, pp 176 to 181.

25. Kirby, P. "DESKTOP Graphics", Datamation, May 1979, pp 164 to 170.

26. Horowitz, E. (Editor) "Practical Strategies for Developing Large Software Systems", Addison-Wesley Publishing Co., Inc., 1975.

27. Myers, G.J. "Reliable Software through Composite Design", Van Nostrand Keinhold Co., 1975.

28. Cho, C.K. "An Introduction to Software Quality Control", John Wiley and Sons, 1980.

29. Dahl, O.J., Dijkstra, E.W. and Hoare, C.A.R. "Structured Programming", Academic Press, 1972.

30. Knuth, D.E. "Structured Programming with GO TO Statements", ACM Computing Surveys, Vol. 6 No. 4, December 1974, pp 261 to 302.

31. Bedhm, B.W. et al — "Characteristics of Software Quality", North-Holland Publishing Co., 1978.

32. Myer, G.J. — "The Art of Software Testing", John Wiley and Sons, 1979.

33. Jacobs, D.A.H. (Editor) — "Numerical Software - Needs and Availability", Academic Press, 1978.

34. — "GINO-F User Manual", Computer Aided Design Centre, Cambridge, 1976.

35. — "Extended Fortran - ICL Student Edition", Technical Publication 4269A, International Computers Ltd., 1974.

36. American National Standard Institute, Inc., (ANSI), "ANSI FORTRAN, X3.9-1966", ANSI, New York, 1966.

37. American National Standard Institute, Inc., (ANSI), "ANSI FORTRAN, X3.9-1978", ANSI, New York, 1978.

38. Balfour, A. and Marwick, D.H. — "Programming in Standard FORTRAN 77", Heinemann Educational Books, 1979.

39. Meissner, L.P. and Organick, E.I. — "FORTRAN 77 featuring Structured Programming", Addison-Wesley Publishing Co., 1980.

40. Bohl, M. — "A Guide for Programmers", Prentice-Hall, Inc., 1978.

41. Bauer, F.L. (Editor) — "Software Engineering - An Advanced Course", Springer-Verlag, 1973.

42. Hughes, J.K. and Michtom, J.I. — "A Structured Approach to Programming", Prentice-Hall, Inc., 1977.

43. Guedj, R.A. and Tucker, H.A. (Editors) — "Methodolody in Computer Graphics", North-Holland Publishing Co., 1979.

44. Barnhill, R.E. and Riesenfeld, R.F. (Editors) — "Computer Aided Geometric Design", Academic Press, 1974.

45. Faux, I.D. and Pratt, M.J. — "Computational Geometry for Design and Manufacture", Ellis Horwood Ltd., 1979.

46. Chasen, S.H. "Geometric Principles and Procedures for Computer Graphic Applications", Prentice-Hall, Inc., 1978.

47. Besant, C.B. "Computer Aided Design and Manufacture", John Wiley and Sons, 1980.

48. Ryder, R.A. "The Engineers' Computer Handbook", Thomas Telford Ltd., 1980.

49. Furman, T.T. (Editor) "The Use of Computers in Engineering Design", The English Universities Press Ltd., 1970.

50. Barsky, B.A. "Computer-Aided Geometric Design - A Bibliography with Keywords and Classified Index", IEEE Computer Graphics and Applications, July 1981, pp 67 to 109.

51. Bliss, F.W. and Hyman, G.M. "Selecting and Implementing a Turnkey Graphic System", IEEE Computer Graphics and Applications, April 1981, pp 55 to 70.

52. Kirkland, W.G. "Cold Roll Forming Practice in the United States", (Paper No. 510), American Iron and Steel Institute, 1959.

53. Martin, S.J. "Numerical Control of Machine Tools", English Universities Press, 1970.

54. Leslie, W.H.P. "Numerical Control Users' Handbook", McGraw-Hill, 1970.

55. Armit, A.P. and Elliot, W.S. "The Design of Interactive Systems for CAD", Proceedings of the 2nd International Conference on Computers in Engineering and Building Design at Imperial College, London (CAD76), IPC Science and Technology Press Ltd., 1976.

56. Day, A.C. "Compatible FORTRAN", Cambridge University Press, 1978.