1/

0-315-26751-8

National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Division
Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulees ou dactylographier

Full Name of Author — Nom complet de l'auteur

ESTOMIH M. S. KOMBE

Date of Birth — Date de naissance

JULY 1954

Country of Birth — Lieu de naissance

TANZANIA

Permanent Address — Résidence fixe

P. O. Box 6,
MWIKA - MOSHI, TANZANIA.

Title of Thesis — Titre de la thèse

A COMPUTER-AIDED INVENTORY CONTROL SYSTEM FOR SMALL MANUFACTURING FIRMS.

University — Université

UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

M. Sc.

Year this degree conferred — Année d'obtention de ce grade

Fall 1983

Name of Supervisor — Nom du directeur de thèse

Dr. T. C. SPRAGUE

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

2nd August 1983

Date

Signature

E Sh Kombe

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

A COMPUTER-AIDED INVENTORY CONTROL SYSTEM FOR SMALL MANUFACTURING FIRMS

by

(C) ESTOMIH M.S. KOMBE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL ENGINEERING

EDMONTON, ALBERTA

FALL 1983

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ESTOMIH M.S. KOMBE

TITLE OF THESIS A COMPUTER-AIDED INVENTORY CONTROL SYSTEM FOR SMALL MANUFACTURING FIRMS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED FALL 1983

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) ...Kombe.....

PERMANENT ADDRESS:

P. O. BOX 6
MWIKA - MOSHI,
TANZANIA.

.....

DATED July 28th 1983

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled A COMPUTER-AIDED INVENTORY CONTROL SYSTEM FOR SMALL MANUFACTURING FIRMS submitted by ESTOMIH M.S. KOMBE in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in MECHANICAL ENGINEERING.

.....

Supervisor

.....

.....

Date..July 13th, 1983.....

ABSTRACT

Recent advances in micro-computer software and hardware technology have brought to the market, computing packages that offer a useful service to industry. This has prompted medium size and even small manufacturers to consider possible achievements in employing these facilities.

This thesis project concentrates on the use of an I micro-computer package (or any similar system), for the Inventory functions of a medium or small sized company like Argo Engineering Limited, of Edmonton. Inventory Identification procedures are developed to provide the necessary tools for the inventory control system.

This project is one of three projects, two M.Sc. thesis and one M.Eng. report, designed to develop an integrated production and inventory control system. Aspects in the eventual assembly of the three subsystem programs are discussed in some detail in chapter six of this report. With slight modification, it should be possible to combine the programs into one package.

Acknowledgements

The author wishes to extend a sincere note of thanks to his supervisor and academic advisor, Professor J. C. Sprague for his assistance and guidance during the preparation of this thesis.

Much of the data used in this report was obtained from Argo Engineering Ltd., through the kind co-operation of Mr. N. Loblick - the president. His support is very much appreciated.

Thanks to each one of those who have directly or indirectly helped in the course of this work.

Table of Contents

LIST OF FIGURES

1. INTRODUCTION

1.1 The Manufacturing Firm

Argo Engineering Ltd. operates two plants, one in Edmonton Alberta and one in Ajax Ontario, with its main office in Edmonton. The company also has a number of local and international manufacturing and marketing arrangements. It specializes in the development and manufacture of material handling equipment, mainly dock levellers, elevating docks, hand & platform trucks and lift tables. Over the years the company has expanded its operations both in the variety of products produced as well as in the number and design of individual products.

This project involves the development of an inventory identification system and an inventory & production scheduling and control system for a manufacturing company like Argo Engineering. It is part of a three-part 'Computer-aided Production Planning and Control System' for such a manufacturing facility. The three areas of study and the students involved are:

1. Quek, Dennis - M.Eng. Report:
   Development of a Computer Program for Material Requirements Planning From End-Product Demand.

2. Lim, Beng - M.Sc. Thesis:
   Demand Forecasting, Master Production Scheduling and

Capacity Planning (Short Term and Long Term).

3. Kombe, Estomih (this report):
   Inventory Identification, Inventory Control and Production Scheduling System.

Figure 1.2 is a diagram showing how the different elements of the Production Control System are linked.

While actual working data and information used in the project is applicable to, and from Argo Engineering, the accompanying discussions and evaluations have maintained the generality essential for such a project.

1.2 Background Information

There are many inputs to the manufacturing process of todays industry. Not only has production grown in volume but also in the complexity of the production of the individual item. Engineering design and development, while improving the quality of the product, has brought with it a multiplication of operations, use of more operations in the process and a need for extra precision. Market forces on the other hand, make it even more important to ensure that production is economical. A high quality product may not offer the success desired by a manufacturing firm if the costs associated with its production necessitate a price level that is not acceptable to the consumer- in absolute or relative terms.

Computer-aided Production Control is on the increase today as manufacturers are faced with the need for an efficient, low cost production as the only means for survival. The versatility of today's computer in data/information storage and processing offers an unparalleled tool for data manipulation and timely decision making. The use of the computer, however, has to be accompanied by accurate and relevant input information, programming and correct interpretation of computer output. Failure to recognise these requirements may result in a production system that is far from economical.

The development of an inventory and production control system requires an effective inventory identification system to be used by the different manufacturing & service departments, dealers and customers, as well as for the data processing system itself.

In view of the aforementioned requirements, the first task is the development of the inventory identification system. Although a good proportion of items used by Argo Engineering in its manufacture are already numbered, lack of total coverage, apparent inconsistency in numbers used by the two main branches, together with the need for a computer aided control system has necessitated a fresh look at the whole question. After reaching an acceptable inventory identification system the focus will then be on inventory control and production scheduling.

1.3 Methodology and Scope

The design of the inventory identification and production control systems will be accomplished with a view to incorporate these and other results into a full production and Inventory Planning system. Thus, the production scheduling and inventory control system will receive inputs from the demand forecasting system and will in turn feed necessary information to the capacity planning system for appropriate allocation and utilization of other resources.

As will be seen in chapter three, there is no single identification system that can be singled out as the 'best' one. One can only identify the best system for a particular type of inventory, number of items, company requirements etc.. The system is thus tailored for unique application. Therefore, in the development of the inventory identification system a compromise between the many factors that have a bearing on the kind of identification system that will best offer the desired results is often necessary. Figure 1.1 depicts some of the considerations in designing a viable identification system.

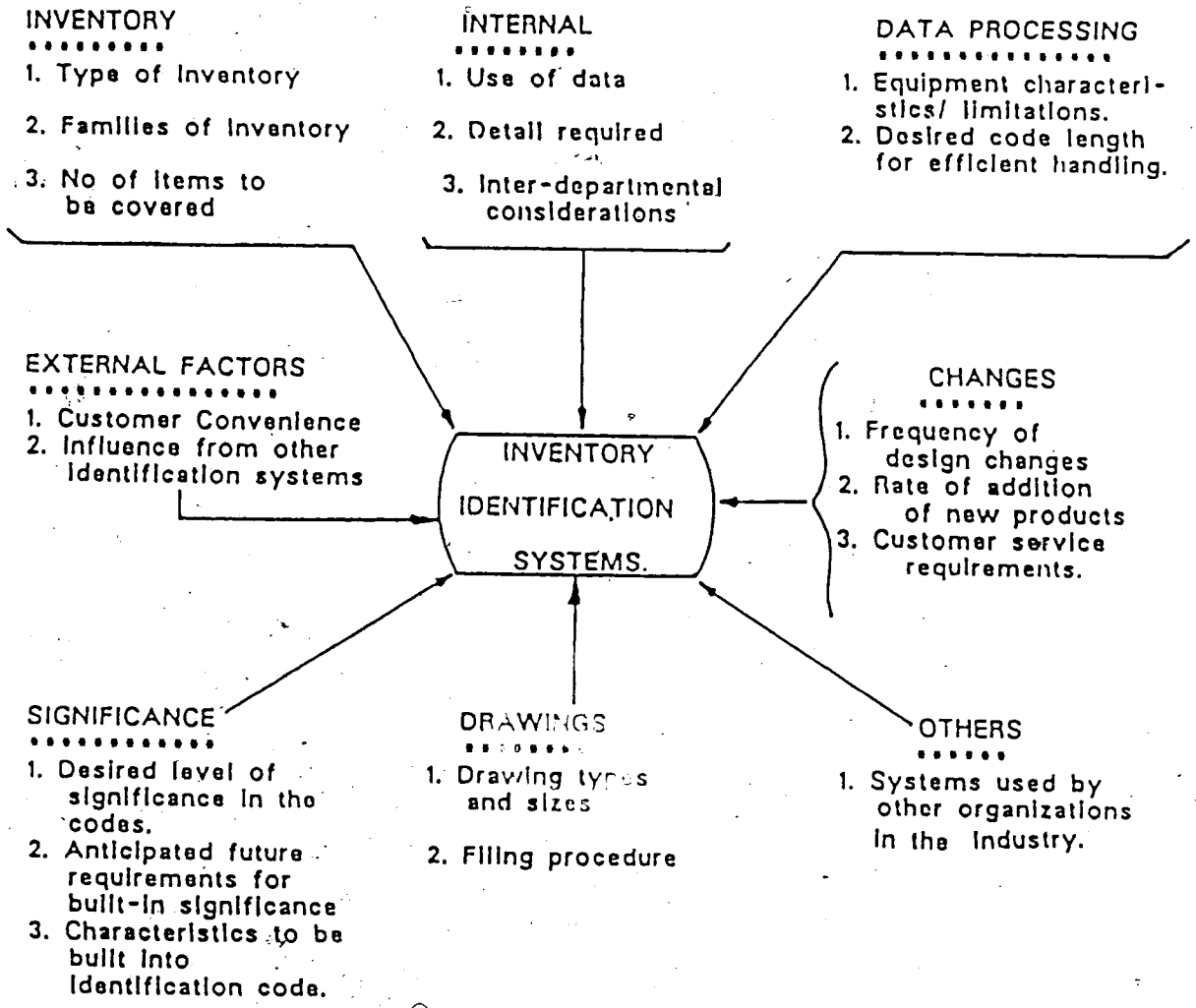


Figure 1.1 Considerations in Designing a Viable Inventory Identification System

Inventory and Production Control

In the development of an Inventory and Production Control System much of the necessary information has to be based on company experience over the years. Amongst the key parameters that need to be established are the following:

1. order quantities,
2. delivery lead times,
3. safety stock level,
4. economical production lots for parts, assemblies or final products as applicable, and
5. planning period (horizon).

In Figure 1.2, the various elements constituting the identification and control system covered in the project are illustrated.

As mentioned earlier and depicted in Figure 1.2, this project is linked to the other two projects in the total Production Control System.

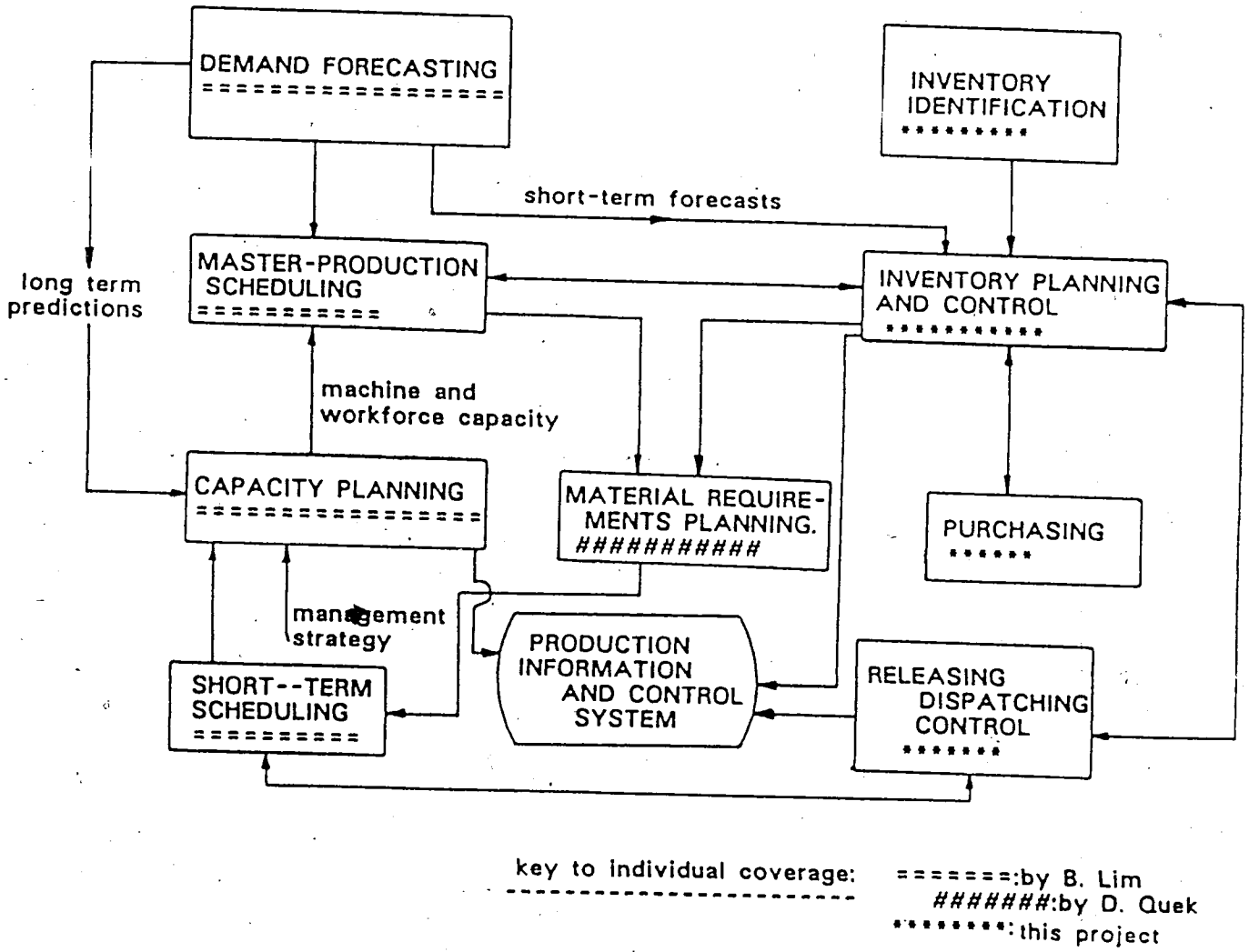


Figure 1.2 Schematic of an Integrated Production and Inventory Control System.

2. PROJECT RESULTS

2.1 Inventory Identification

The task of establishing a viable inventory identification system is central to the success of any computer-aided system. In most cases, when consideration is given to computer control, the company's inventory identification procedures are inadequate to support the new system. Either these procedures are just informal, or if formal, they do not meet the necessary requirements for computer data processing.

An inventory identification system, which is based on the inventory type found at Argo Engineering Ltd. has been developed, and is presented in this report. The system is based on a combination of both significant codes and non-significant codes, noting that each one of the two has unique advantages. Although the identification system is tailored for a specific company's inventory, only product re-assignments to one of the codes(the product code) should be necessary in order to use the system for a company with different types of products.

2.2 Inventory & Production Control

The following programs have been developed to address the inventory control problem.

1. A general transactions program, incorporating receipts and deliveries, alterations, order processing, sorting,

exception reporting etc..

2. A parameter evaluation and regular report generation program also incorporating the exception reporting subroutine.
3. An inventory identification program, - with access to the inventory sorting subroutine.

The programs are developed in the form of a number of different subroutines to optimize on memory capacity. Should some modifications to the programs be necessary, only the applicable file out of the many files used for the programs needs to be changed. When this has been done the changed program file will have to be re-compiled and the run file obtained by re-linking the object files. The different program units are illustrated in Figure 4.1.

The control system programs have direct access reading and writing capability, which is very convenient for the kind of transactions performed on such a system. Information sharing between program units is achieved by the use of "common blocks", which makes it unnecessary for each unit to have unique storage allocation for parameter values that are common to other program units. Common blocks are established in the programs by declaring, at the beginning of each applicable program unit, those sets of variables that are common to other program units. Only one memory allocation is made for any one of such data, which cuts the total memory requirement considerably.

2.3 The Integrated Control System

It is intended that the programs developed in this project be integrated with other programs that centre on different areas of the production control system as outlined in the abstract. Chapter six of this report concentrates on the possible approaches in integrating the different programs, as well as on the capability of the IBM package in handling the programs. It is noted that with slight modification the programs can be assembled following either one of the approaches described in chapter six, as conditions may dictate.

3. INVENTORY IDENTIFICATION

3.1 Introduction

Inventory identification in industry, normally aided by coding and classification, is aimed at assisting the many functions of manufacturing by providing a logical and meaningful system of identification for information, data and components. Without a proper system of parts/component identification, a communication breakdown within the organization and between consumers and the supplier usually results, and the process of manufacture and information storage and retrieval develops to a state of confusion.

The development of meaningful and systematic identification classifications and codes has been around for several decades but its application has been basically confined to large organizations and industries. Growth and diversification within industries, together with more economical computing facilities at present justify modern production control for the medium and even the small sized company.

3.1.1 Classification and Coding

Classification may be defined as *'a systematic arrangement of similar items into suitably selected categories'* [12]. Classification aids in identification since it makes it possible to develop (through coding) a means of identifying

a particular inventory code with a corresponding inventory class. To avoid ambiguity it is important that the classification categories are mutually exclusive and do not separate parts or products that are closely linked. Coding on the other hand *"is the allocation of symbols (alpha numeric etc.) to the classification categories"*[12] mentioned above. An identification system would normally have as many codes as there are classification systems. Apart from the significant codes identified in the next section, which convey particular information about the inventory item, there are normally additional serial numbers to further differentiate between individual items. It is only with fully significant identification systems, which are quite rare, that these additional characters are not required. In a fully significant identification system each character in the identification code will be designed to convey specific information about the inventory item in question. Some definitions of some of the terms used is given in section 3.2 on identification approaches.

The development of an industrial classification system is a compromise between the many demands made on it by its users. One, therefore, tries to satisfy as many of those demands as possible while observing the limitations inherent on the identification system. Classifications must consider the following demands amongst others:

1. Essential needs of the users must be established and a compromise reached.

2. Only permanent characteristics contained within the information, data or components should be selected and used in the design of the classifications.
3. The definition of the classification categories must be precise and unambiguous.
4. The classification must be comprehensive, the categories being capable of including all that comes within the classification coverage (i.e. the scope of the design).

As partly evident from the introduction , once established, the identification system serves three major functions or purposes. These are also the key areas of study in the process of developing the system since it must satisfy 'their' expectations. These functions are:

1. Internal use for identification and reference of items/parts in the process of planning and production within the different departments in the organization. Word descriptions are too long and inexact.
2. Customer facility in the process of ordering or buying from a manufacturing firm or supplier. Easy and clear identifications eliminate confusion that may cause shipping or producing the wrong item.
3. Computerised Production and Inventory Planning and Control (PIPC) systems need a way by which one can transfer items from bills of materials into specific materials and parts needs. Numbers/codes are short, exact and much more unique. Built-in significant codes

are useful when information has to be input or output for a particular group of items that have something in common which should already be built into the system. When there is no means of specifying the particular group of inventory then the only alternative is to process one item after another.

An identification system in a manufacturing industry is normally involved with one or more of the following inventory classes:

1. raw materials,
2. work-in-process and
3. finished goods.

The degree to which these classes are covered depends on such factors as: the requirements from the three functions cited above, the manner/ process of production (for work-in-process), and the extent of computerization of production functions. Normally components and assemblies bought from outside are considered as raw materials though if more convenient they may be classified separately.

Raw materials can be described as those items purchased to be further processed. They may be from outside or from another division within the organization.

Work-in-process inventories are raw materials that have had labour and additional costs added and are awaiting further processing into finished products.

Finished Goods are those products at the end of the production sequence, available for delivery. They may be

carried in inventory or shipped upon completion.

Overlap of Work-in-Process and Finished Goods.

Although the above description indicates clear distinction between finished goods and work-in-process (sub-assemblies, assemblies etc.), one notes that in a number of cases requirements dictate that work-in-process be shipped to a customer. This happens mainly when customers need replacement units or when some outside manufacturer places an order for component parts. When such is the situation the tendency is that one would like to consider such a job also as a finished product. It is thus not appropriate to consider the mere fact that an item is ready for shipment as an exclusive implication of a finished product.

3.2 Basic Approaches

Individual identification systems developed by different users are numerous. It is not possible to enumerate all such possibilities. Despite individual differences in detail and composure, there are still many similarities between the many identification systems in use. This section offers, quite briefly, the basic characteristics of these identification systems.

Some Definitions:

Before going further into the description of identification systems, it is necessary to list some of the terms that will be employed, as defined by Wallace [19] and Johnson.[9]

Part Number;

A number which serves to uniquely identify a component, product, or raw material. Sometimes referred to as stock code or product code.

Significant Part Number;

Part numbers that are intended to convey certain information such as the source of the part, the material in the part, the shape of the part etc.

Non-Significant Part Number;

Part numbers that are assigned to each part but do not convey any information about the part. They are mere identifiers not descriptors.

Semi-Significant Part Numbers;

Part numbers in which only some of the fields or digits represent significant information about the part. The rest of the fields are added only to achieve unique identification (non- significant).

2.1 Types of Identification Systems

Inventories are basically identified by one of the following: 1. numeric characters, 2. alphabetic characters, 3. alpha-numeric characters, or one of the three with some form of punctuation with dashes (-) or similar continuation characters.

All Numeric System.

Parts identification systems or symbols that include only numeric characters are accordingly referred to as all-numeric systems. Where desired dashes are added to separate basic classifications. Examples include the following:

"146788" - a Single set of numbers, and

"2345-122" - a semi-significant system with a broken set of numbers.

The amount of significance built into the system will depend on the particular use of the system. An identification system purely for design retrieval purposes may have a single set of numbers each one coded according to one of several descriptive sections.

All-alphabetic code systems.

This category of parts identification symbols are rarely encountered in practice, mainly because of the general familiarity with the numbers which makes them much more useful. This is particularly so when one has, within the system, a set of non significant characters merely for

unique identification.

Although all-alphabetic code systems are not common, alphabetic characters are frequently used together with numerical characters, as will be discussed under alpha-numeric code systems. The use of alphabetic characters for some significant codes offer a number of advantages. First, the range of characters available, twenty six of them, provides for a large number of classification categories. This is very useful when more classification categories are required than can be offered by the ten numeric digits. Another advantage of alphabetic characters is the ease of association derived from a matching of item names(eg. the first letters of the names) with the classification characters used. When the letters used for the significant codes correspond to, or relate to letters in the names of the classification categories, it makes it easy for one to decode the significant codes.

Alpha-Numeric Systems.

This category of parts identification symbols include both numeric digits as well as alphabetic characters, intermingled, often coded. The complexity (or simplicity) of the code depends on many variables including the type of product, its mechanical complexity and the needs of the consumers and producers as to the degree of significance to be built into the identification system. Examples include the following:

12T-24Z family type symbol,

1A2-6B9-7D8 functional significant system,

A-54312-B drawing number file system, and

A1B2F6C4 alternate character system.

To some extent, particularly when some degree of significance is desired, alpha-numeric systems may be quite ideal, combining the benefits of both character types.

Numeric characters, particularly when used in the non-significant portion of the code, provide a wide coverage, i.e. the number of inventory items that can be accomodated by the system is enlarged. These characters are also useful for significant codes where ten or less classification categories are required, and where specific character association is not important. Alphabetic characters in the identification codes provide benefits as discussed under alphabetic code systems. The combination of the two character types, thus, results in a much more effective and efficient identification system.

Any single identification system will have its own advantages and disadvantages. It is the kind of demands placed on a system by its users that enables some of the advantages of a particular system to be emphasized while some shortcomings are overlooked.

In the course of designing a particular identification system numerous questions will be raised. Depending on the particular situation some of the following considerations should prove very useful.

1. Does the system provide convenient filing characteristics for the different departments?
2. Is the system compatible with conventional data processing equipment?
3. Is the system easy to administer?
4. Is the system acceptable to each of the departments?
5. Are there possibilities of misapplication and confusion during use?
6. Does it provide for derivation of engineering drawing numbers?
7. Is it suitable for centralised or decentralised control?
8. Does it offer good identification characteristics?
9. Will it easily accomodate future design revision, expansions etc?
10. Is there a requirement for special handling skills?

The following section concentrates on the particular inventory type and identification requirements for Argo Engineering Ltd., and proceeds to formulate a viable strategy for the assignment of inventory code numbers to the inventory items in question.

3.3 Company Requirements

Before proceeding to the development of an inventory identification system one must clearly establish the basic requirements. The major requirements to be placed on the new system must of necessity be identified by the company. A discussion of the major functions that company officials place high amongst other requirements is thus presented below.

1. The system should provide for an effective means of locating necessary information about a specific item used by the organization. This point refers mainly to parts that have engineering drawings. Thus, the identification system should offer a quick reference for the employee who wants to locate this particular item and/or drawing.

2. The system should readily provide information regarding the basic category of inventory in which a particular item falls. For instance, someone dispatching work orders should be able to distinguish between those orders going to the raw materials warehouse and those directed to the assembly shop. Alternatively or in addition, it should be possible to associate the identification number of an item with the type of product it is used for or the family of inventory they constitute, like electrical items, hydraulic components etc.

3. It should be possible to use the system to retrieve

information about like items. For example, this information may be required when a new identification number is about to be assigned or when a substitute item is being sought. In the former case one needs to ensure that this particular piece of inventory is not already stocked. One, thus, avoids duplication of identification numbers. In the latter case one is interested in inspecting those items with somehow similar characteristics to an item that is not in stock at the moment. The objective is to get an item that can be used as a substitute, if need be, after some modification.

In both situations above, assuming a computerised data storage system, which should eventually be the case, codes in the identification system are required that will enable one to perform a computer search.

3.4 Towards an Appropriate Identification System

Having pointed out some of the key requirements of the system to be developed in the preceding sections, this section is devoted to a step by step analysis of these and other functional and limiting considerations. The final subsection in this section culminates in the development of the proposed identification system and a definition of terms as applicable to the classification terminology.

3.4.1 Nature of Inventory Under Consideration.

While an identification system must allow for expansion of company activities, thus being as flexible as possible,

it is indeed a matter of practicability that the system must nonetheless be tailored for the particular manufacturing system. In the manufacture of any particular product, materials, parts etc. enter the process at different stages or levels. This can be explained by the use of the bill of materials figure below. Reference to high level and low level inventory will be made accordingly. (This terminology is used in material requirements planning systems)

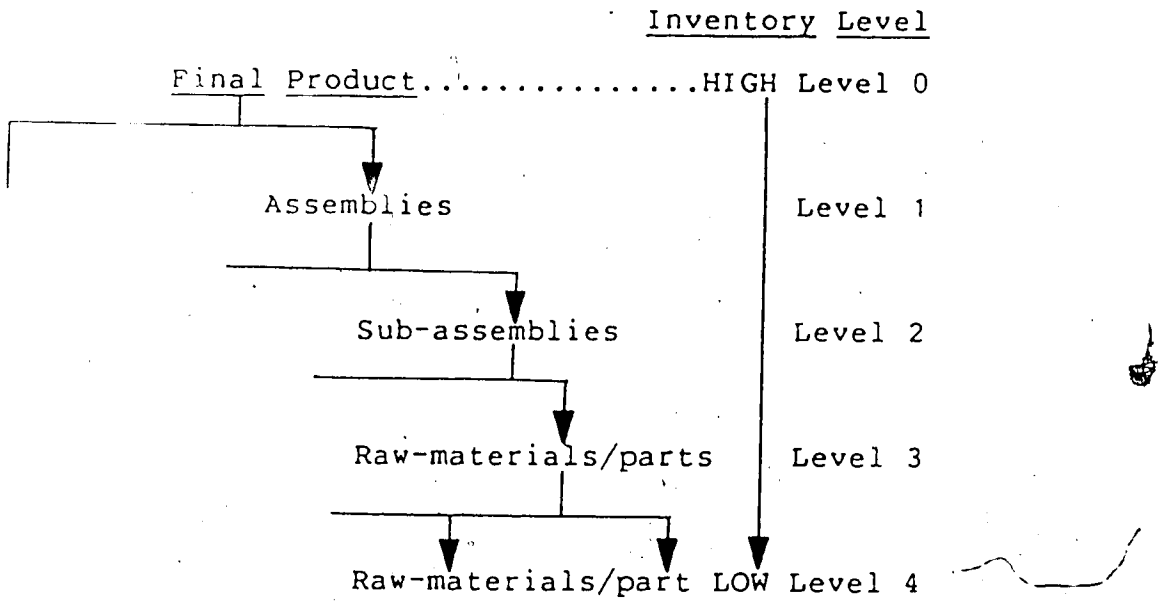


Figure 3.1 Inventory Level Illustration.

The types of inventory applicable to Argo Engineering Limited are:

Finished Products, which will include the following: dock levellers, elevating docks, trucks (platform trucks,

hand trucks etc.), cranes, dock boards and plates, conveyors and ladders.

Work-in-process inventory will include various forms of sub-assemblies and assemblies at different stages of manufacture. For example, they may be fabrications or machined components.

Raw Material Inventory is mainly composed of metal sheets and bars (noting the amount of fabrication involved) of various sizes and shapes. Included in this category are also items purchased from outside to be incorporated in the manufacturing process at different stages. These are standard items like nuts and bolts, as well as special equipment like motors. A considerable number of items at the lower levels are used for more than one assembly or product. In these cases one needs to avoid having several identification numbers for the same part.

The basis usually used for coding raw materials by the kind of material itself can not be used with finished parts and components because these are often whole assembled products. These products may be pumps, motors, compressors etc., none of which can be classified according to the kind of material it is made from. Thus the use of kinds of material would give different codes to functionally similar items of different material.

Another approach could be to use a coding that allocates similar codes to different kinds of items, eg. shafts, gears, housings etc. Each of these items would thus

have its own type of code, regardless of the material it is made of, or the product it is used for. This approach, while distinguishing between these different items, has the disadvantage of assigning parts of a finished product numerous unlike identification numbers.

It appears more favourable to base the significant codes on the types of products in which the components are used and other general characteristics, rather than specific material or item types. At the same time, items that are used for specific functions on various assemblies and products may be identified according to these special tasks.

3.4.2 Data Processing Requirements.

Developments in the area of computer and/or micro-computer technology over the past few years have eliminated most limitations on the use of numeric and alphabetic characters. At present, highly versatile computer systems with high processing speeds and significant storage facilities are available. Our choice of characters to be used in the coding should therefore be mainly based on other considerations, that is, other than limitations on data processing equipment.

3.5 Classification and Identification Procedure

Preceding sections have dwelt on the many useful characteristics of an identification system, requirements for good performance, limitations, special requirements for the company in question etc.. This information was directed towards assisting in designing the kind of identification system that will suit this type of industry.

Some amount of significance has been built into the item codes to offer basic information and characteristics about the items/components. This will be very useful whenever one has to search for items of some predetermined features/ types. By using a combination of these codes one can narrow the number of items that should be considered. Further details about the individual items will normally be available from system files in storage. It wouldn't make sense therefore, given the present requirements, to build such detail into the inventory identification itself.

Four significant codes are proposed to be built into the inventory identification system for Argo Engineering Limited. These significant codes may constitute either the first four characters in the identification code or the first three and the last one, whichever appears more appealing. As will be explained later, the first arrangement proves to be more suitable.

The significant codes will provide identification as follows:

1. Final product or special inventory code, to be an

alphabetic character.

2. Inventory level/production process code, to be a numeric character (0-9)
3. Geometric form of inventory code, as and if applicable, also to be a numeric character (0-9)
4. Engineering Drawing size/location code, an alphabetic character.

The following sub-sections describe the above four codes in more detail, together with the additional numeric characters to be added to the four above in order to offer the desired unique identification. Should the need arise, one of the four additional characters may be used as a check digit for guarding against faulty item numbers being input to the system.

3.5.1 Final Product or Special Inventory Class Code

As the sub-heading indicates this code describes an item either by the particular end product it is used for (or mostly used for), or by the special class of inventory in which the particular item belongs. The use of special class identification is an attempt to avoid a situation where many items used for a variety of end products (eg electrical components) have to be identified by one end product. Thus, instead of identifying, say a motor with a dock leveller when it is in fact used on many other products, it is identified under 'Electrical'. The same system applies to

other electrical items. This does not mean however, that one can't have items that are used for more than one product but identified with only one product. What will be achieved is a high reduction in the number of such cases. The important thing to be observed, is to assign only one identification to each inventory item regardless of the number of end-products in which the item is used.

Because of the number of groups involved in this classification together with the ease associated with matching many alphabetic characters with the groups they represent, alphabetic characters are considered ideal for this code.

One character or a set of more than one character (say 2) could be employed to achieve this code. The use of two or more alphabetic characters to identify a final product has the advantage of allowing for many combinations of letters but increases the total number of characters in the final code as well as makes it more difficult to match the letters with the corresponding product. A number of alphabetic characters are still available for use with new products, and one may assign one letter to more than one product (in the event of excessive future needs), to avoid having too many product codes. In view of this factor, it appears more appropriate to use a single character for this code.

The code characters selected and the groups they represent are as follows:

Elevating Docks.....A

Dock Boards, Dock Plates.....B

Conveyors.....C

Dock Levellers.....D

Electrical.....E

Forklifts and Attachments, Pallet Trucks.....F

Hydraulics and Pneumatics.....H

Cranes and Monorails.....K

Ladders.....L

Hand Trucks.....N

Platform Trucks and Trucks General.....P

Racking and Shelving.....R

Special Equipment (not in other groups).....S

Tools.....T

Standard/ Commercials.....X

The above letters represent the first character of the inventory identification code.

In deciding in which of the above groups a particular component should be classified, one should first see whether it belongs to any of the special groups, ie. the standard items, tools, electrical components or the hydraulics and pneumatics. If it does not belong to any of these groups then it should be coded to correspond to the final product representing maximum use. Should it be difficult to associate the particular item with any of the final products then it should be classified under 'S' to show that it is

either a special item or that it just doesn't fit well i any of the other classes.

3.5.2 Inventory Level and Production Process Code.

Apart from getting information related to the product an item is used for, or the special category it belongs to, it is very useful to be able to identify the level of inventory in question. Many industries have different storage areas for different levels of inventory, which makes such identification very relevant. Bulk raw materials and single item raw materials may also be maintained separately, and so can sub-assemblies as opposed to assemblies and/or finished products. This code also finds use when searches ... items of some predetermined characteristics are to be e. Coupled with the production process code for individual components (single items), and supplemented by codes for the other classifications, one is able to narrow the choices to just a few items for manual inspection.

The inventory level and production process code occupies the second position of the inventory identification code. The breakdown of these codes (to be achieved by the digits 0-9) is as follows:

Raw materials, unprocessed.....0

Raw materials, single items.....1

Raw materials, other than 0 or 1 above.....2

Single component, cast and/or machined.....3

Single item, forged or formed.....4

Individual item, welded.....5

Individual item, by other process/operation.....6

Sub-assembly.....7

Assembly.....8

Finished Product.....9

Definitions:

The following is a definition of the above terms as they will be applied in the choices of codes.

Raw Material: All material/items from outside the company on which no work or transformation has been done. It should not matter in what form or level they enter the production line. Examples are sheet metal, standard bolts and nuts, paint, electric wire, where all are purchased material.

Unprocessed: Used in relation to raw materials above to imply that it is either in bulk or in a form that can not be used before cutting, machining etc..

Processed Raw Material: Raw material, normally a purchased item that is a distinct single part or assembly (e.g. a purchased spring, bolt, washer, valve or shaft).

Single /Individual item: Used to signify that no two or more parts of a lower level, each with an identification number

constitute the particular component. Examples could be the same as example items above but in this case made within the organization.

Cast, machined, forged, formed, welded etc., have their normal manufacturing process meaning.

Sub-assembly: More than one distinct item assembled but still requiring several processes, additional parts etc. before going to the final assembly.

Assembly: This is an assembly of components at a stage just prior to the final assembly stage. It is the final stage of sub-assemblies.

As examples, the piston, piston rod, and rings put together is a sub-assembly whereas the whole hydraulic unit would be an assembly. The latter goes straight into the final assembly of the product.

A fabrication at its various levels will remain a sub-assembly until at a stage where it is clearly a distinct major unit on its own with respect to the functioning of the end product.

Note:

One product (large & complex enough) will normally be comprised of a number of sub assemblies. During the final assembly of the end product one does not refer to the linking of one assembly with another as a new assembly

unless the nature of the production process calls for the assignment of a distinct identification number to this particular combination. In most cases a distinct identification number will not be necessary.

3.5.3 Geometric Form of Inventory Code

This code is intended to offer a picture of what the item in question looks like, with regard to general features. It does not have anything to do with the material or size of the item. It is much more useful for the raw materials and single items and less useful or less applicable to the complicated sub-assemblies and assemblies. Since this is merely a general description of the item, there will be no harm in using the code, say for 'cylindrical' components (i.e. codes 3,4,5) for a part or sub-assembly that consists of some minor non-cylindrical parts. For example a bolt should be coded 'cylindrical' despite having a hexagonal head, and so should other parts or assemblies in a similar situation. As long as the coded geometric shape is predominant in the component, one should go ahead and use this particular code. Since the second character on the item identification will indicate that it is a sub assembly etc. one would normally expect that it would not normally be wholly 'cylindrical'.

The following are the suggested characters and their corresponding descriptions for this code (0 - 9), which will occupy the third position of the inventory identification

characters.

Plate or Sheet material.....0

Non cylindrical & hollow.....1

Non cylindrical, other than hollow.....2

Cylindrical, hollow.....3

Cylindrical, threaded. 4

Cylindrical, other than above.....5

-blank-.....6

Solid, above descriptions not applicable.....7

-blank-.....8

Fluid.....9

At the moment positions 6 and 8 are left 'blank' to be made use of whenever the need for additional descriptions arise. Finished products should normally be identified with '7', as the third identification character as given above, unless the item is simple enough for one of the other descriptions to be used without loss of generality.

3.5.4 Drawing Codes

Engineering drawings are normally filed according to the drawing size. Small drawings would not normally be mixed with large ones. This means that for two or more consecutive part numbers with parts' drawings of different sizes, their locations will be different. This calls for some means of determining the size (and thus the location) for a

particular item's drawing to be able to locate the drawing. One option would be to keep such information in storage and resort to system retrieval whenever a drawing is required. This method would be both cumbersome and unreliable noting the number of staff normally going after drawings and the amount of time that would be spent seeking such information from the data base. It is necessary to have a simple but definitive means for locating and filing drawings.

It is therefore a worthy undertaking to incorporate as part of the item identification the Engineering drawing specifications. Where a drawing does not exist there should also be a code to give such information to avoid searching for unavailable drawings. It is proposed that alphabetic characters be used for this particular code. This is partly as a means to separate the significant part of the identification symbol from the rest of the characters, as well as in compliance with drawing designations currently in use.

Assuming five drawing sizes (with an A for the smallest size, B for the next larger size, C, D, & E accordingly) , the drawing codes illustrated in figure 3.2 will be obtained. P is used for 'pending', (i.e. a drawing is to be available eventually) and an N for "no drawing available". This code character will preferably be on the 4th position of the item identification.

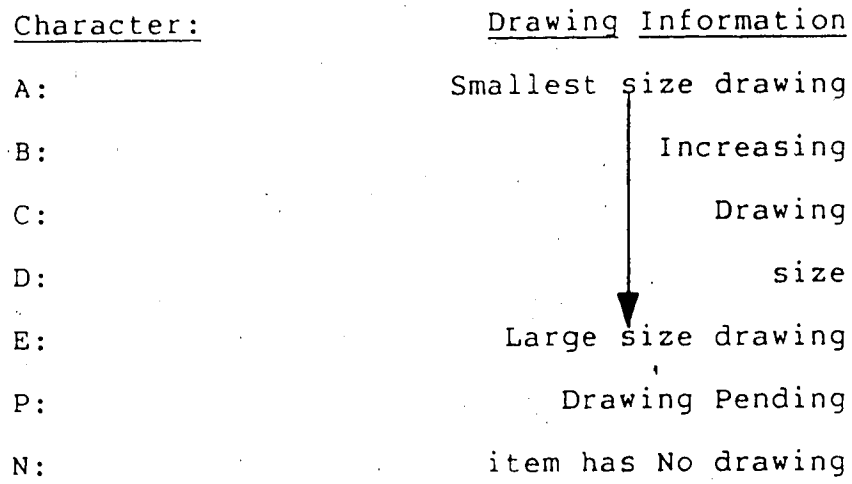


Figure 3.2 **Drawing Code Illustration**

As mentioned earlier, the drawing code character could be placed at the last position of the identification, which may be considered more 'visible'. This however, will mean having the two significant numeric characters immediately followed by the other four characters. It would seem more favourable to place the drawing specification letter before the non-significant digits particularly to avoid attaching significance to the wrong characters in the symbol.

3.5.5 Non-Significant Characters.

Apart from the preceding 4 characters (significant) in the proposed identification code, the next four 'digits' are

non-significant. This means that they do not offer particular information about the item. They only serve as part of the serial characters to distinguish between individual items. The last of the non-significant characters could possibly be used as a check digit. Since we are dealing with many parts and we only have four significant characters many items will bear the same first four characters. The next characters (digits) will be assigned serially, placing like items close together, to offer the necessary uniqueness in the identification.

The allocation of the non-significant numbers should be done in a way so as to put similar items as close together as possible within their different 'significant character' combinations. This will be advantageous during operation, for instance, when looking for substitution items. Unallocated space should be left between each short series of serial numbers for insertion of future design additions.

3.5.6 Operational Considerations

The use of a check digit (according to the illustration in figure 3.3), should be weighted against the need of adding a ninth character to cover the identification, if not now, in the near future. It may happen that one is able to use only seven characters to cover the inventory identification (with three non-significant characters only), in which case one will have the option to drop the eighth

character or adopt a check digit. The usefulness of a check digit depends on several factors originating from the production and inventory planning and control system. If the system is such that an error in punching the identification code for an item will result in unbearable confusion then one may need such a built-in check system. If, on the other hand, one only has a few transactions per day, it is easier to make a routine countercheck of the entries before they are acted upon, in which case one would rarely need a system check procedure. If however, the computer control system was fully integrated across departments and inputs in one department cause a chain of reactions in other departments, such a routine check would not be effective. As the risk increases, the need for a check by the system itself becomes increasingly pronounced.

Check Digit Illustration.

Figure 3.3 illustrates how a check digit is used to detect input errors.

The last digit in the item code, which is a "5", was calculated as per case 1 of figure 3.3, thus the result 144 calculated from the wrong entry (i.e. by case 2 in figure 3.3) will cause an error message from the system, thus prompting the user to check the entry.

Although the check system would not eliminate the chance of a wrong entry going undetected, the chance of such an occurrence becomes minimal and can be made negligible by the

INVENTORY CODE			WRONG ENTRY		
A345B8965			A345B9865		
case 1			case 2		
character	factor	product	character	factor	product
3	1	3	3	1	3
4	2	8	4	2	8
5	3	15	5	3	15
8	4	32	9	4	36
9	5	45	8	5	40
6	7	42	6	7	42
		145			144

Figure 3.3 *Example on the use of a check digit.*

use of an even more elaborate system, such as the use of two or more verification procedures each one with different multiplication factors.

Non-Item drawings

A few special drawings exist within Argo Engineering and most other companies that represent, not an item but a concept, an illustration, or the means to do something. These drawings could be electrical circuit diagrams, an

assembly illustration, an installation pit drawing etc.. There would be no matching item numbers for these items. Neither should any attempt be made to specify unique numbers/codes for these items since they do not constitute inventory.

Drawings of the type above should bear the reference of the particular items they are associated with, in relation to their function. An electric circuit can bear the identification of the item for which the circuit is intended, a pit bears the code for the item installed in it etc. The drawings should be filed along, or in accordance with their size (substituting the actual size for the size code used in the parent item). To differentiate these drawings from the main item drawings one could use -1, -2 etc. as a prefex to the main identification/ drawing code. References to these drawings should be made accordingly.

Filing of Drawings

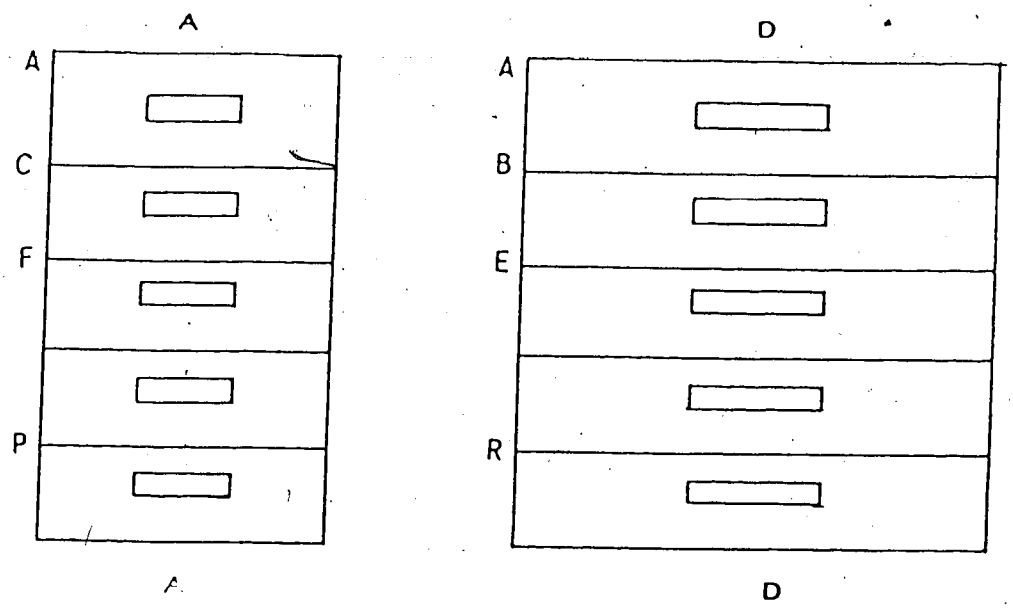
To be able to make maximum use of the identification range available under the described system, the following proposals and guidelines are recommended with respect to the identification of parts and drawings, and the filing of the latter:

1. The item identifications should be used directly for purposes of drawing numbers without any alterations or deletions.

2. Item identification will be achieved by all seven or eight characters used and not merely by the three or four non-significant characters. This method is crucial in enlarging the total number of parts that can be handled by the system without having to employ extra digits. Thus, similar non-significant digits may be assigned to different items as long as their significant character sets are different.

3. Should a change of drawing size be made for a particular item in which case the new drawing necessitates a change in item code (due to a different drawing code), the change should be done while ensuring that this does not cause a duplication of identification. Where necessary other digits may have to be altered to retain uniqueness of the item in question. This should also be followed whenever there is any cause for changing item codes, eg. when a company starts production of an item that was originally bought from outside (raw material).

At the beginning of the system development we may avoid using the drawing codes for distinguishing between items, depending on the overall coverage offered by the other codes. Although drawings are revised periodically, they are normally redrawn on the same size of sheet, unless additional detail becomes desirable. There should, therefore, be no confusion or much renumbering if the



Key: Characters to the left of drawers represent the first character of ID code. (A,C,F,H,L..X)
Middle characters, (A & D) represent the drawing size-i.e. the second alphabetic character.

Figure 3.4 Illustration of Drawing Filing System

drawing codes are used as part of the unique identifier.

In filing drawings, one should keep similar sizes together, labelling the drawers accordingly. The first code character should appear to the left of the drawer, and all items bearing this character as the first character for the particular drawing size should be filed in this drawer.

The location of a drawing for the part number **D42A1404** will be in the left set of drawers (middle A-size), in the second drawer from the top (ie between C and F) and in the numerical order of the number 421404 (within the drawings having 'D' as the first code character). First, the drawing size code is determined from the fourth identification character. This code being an 'A' implies use of the left set of drawers (figure 3.4), where the label 'A' at the top and bottom of the drawer indicate drawing size. The first character of the identification, which is a 'D' in this case, corresponds to the characters appearing in alphabetical order on the left edge of the drawers. Thus, one notes that 'D' should be between 'C' and 'F' which appear on this set of drawers. The exact position of the drawing for the above item is then determined in accordance to the remaining code characters, ie. "42" and "1404". This system may appear lengthy to start with, but should prove to be quite effective following a brief period of use.

A flow diagram for the inventory identification procedure is given in Figure 3.5.

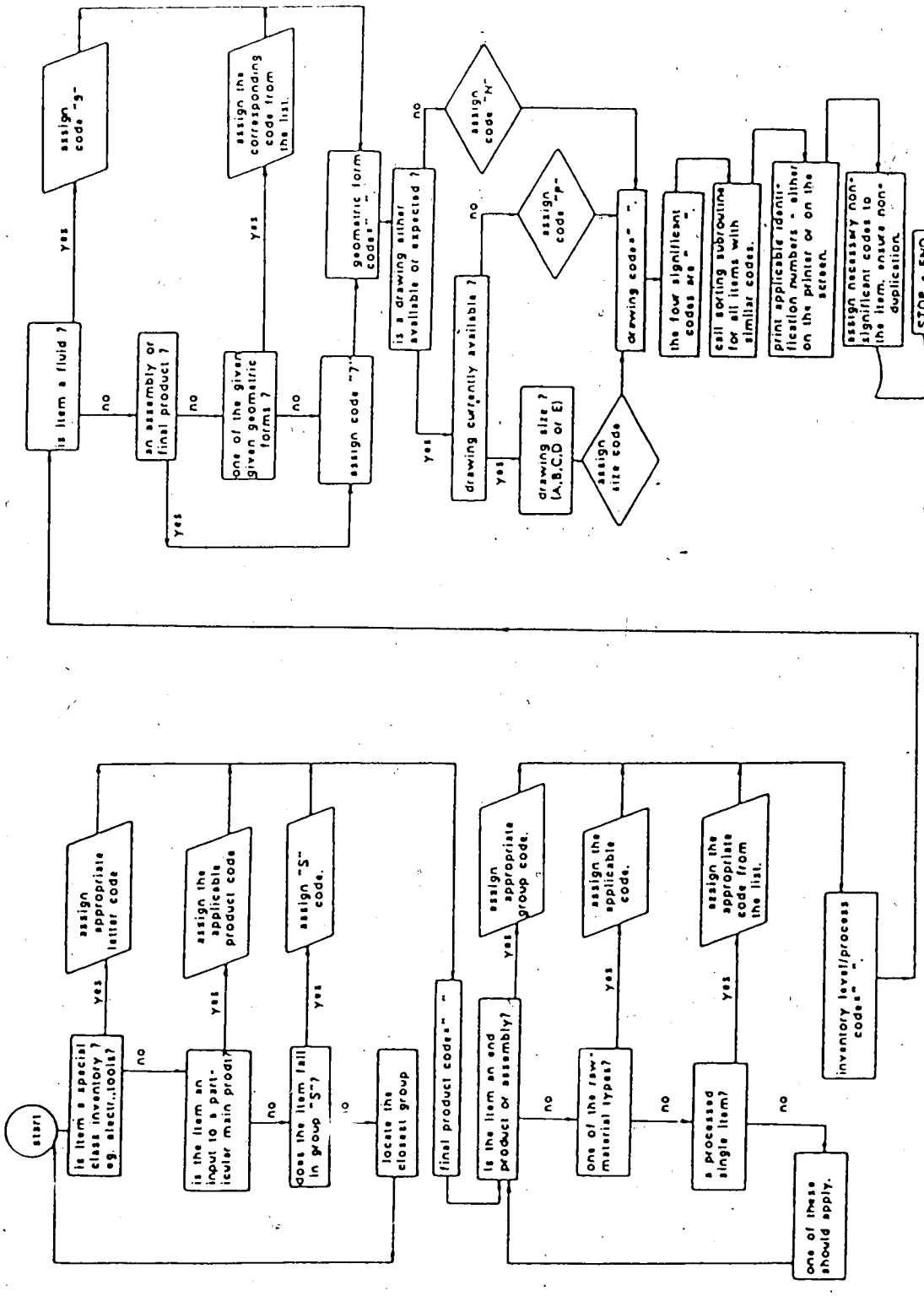


Figure 3.5 Flow Diagram for Inventory Identification Procedure

3.6 Manufactured Products Records

Even after a particular product is purchased from the company there is still need in the future to be able to get information pertaining to component designs used at the time of its manufacture. There should be a means of determining when the product for which spare parts are sought was produced.

In some cases the customer may not even indicate (or may not know) the particular model that was purchased. This situation will arise where a number of different models were manufactured for a specific piece of equipment. In order to get the information required above it is necessary to have the following:

1. A means by which the customer can easily identify a product, which involves having a unique specification that will be used during communications with the company. A serial number plate on each product would no doubt be ideal for this purpose. For example, the plate should read, 'Dock Leveller- Srl #-----', or 'Elev. Dock #-----'. This should be clearly displayed on the machine. Purchase documents given to customers at the time of purchase should remind them to quote serial numbers during any communication.
2. Within component records, information regarding design changes for each individual component should be maintained (this will be discussed further under inventory control: data and information). This should

indicate the time the change was made and the serial number of the last product before the change. Thus, when a service job is received, one can use the quoted serial number to check the time of manufacture of the product and subsequently the parts that were in use at the time of its manufacture.

4. THE INVENTORY CONTROL SYSTEM

4.1 Introduction

The problem of Inventory Control in a manufacturing organization does not find a final solution from a material requirements planning system (MRP)-[3]. There are several reasons for this. First, the fact that MRP enables a company to foresee its inventory requirements in the future does not eliminate possible variations in demand that require a level of flexibility in inventory availability. Such flexibility could possibly be achieved by expediting orders, which turns out to be very expensive most of the time. Flexibility can otherwise be accommodated by allowing some 'safety stock' in inventory. The same applies with regard to other variable factors like procurement lead times and scrap allowances whose variation can cause inventory to be depleted, causing manufacturing stoppage.

The second reason has to do with economic considerations in placing orders. Quite often it is necessary to vary the order quantities resulting from the MRP explosion. Strict adherence to the MRP figures would otherwise result in uneconomic orders. The same is the case with regard to in-house production quantities. The MRP system, for instance, can indicate that 5 pieces shall be needed by date 15, but where the economic production quantity is 10 pieces. This means that 5 extra pieces have to be carried in inventory when the economic lot is

produced. One can therefore, say that the MRP system performs so as to ensure that the minimum necessary purchase and/or production quantities are made available. Whether these minimum quantities are ordered or not is a question that is handled by the inventory control system.

Apart from providing the necessary tools for meeting MRP requirements, the inventory control system is the medium for all inventory transactions and status reports. A major program, included in the appendix, was written to handle the anticipated transactions, as well as generate the necessary system reports. Additional subroutines are included to perform a variety of applications useful for management decision making in this important area. The following sections of this chapter will be devoted to the different aspects considered in the process of designing viable inventory control policies.

4.2 Independent and Dependent Demands

The concept of independent and dependent demand is the basis of MRP systems[3]. MRP systems are designed to handle dependent demands, i.e., the demands that are a result of higher level inventory requirements (inventory levels are described in section 3.4.1). Within companies like Argo Engineering Ltd., however, one also experiences independent demands at the various levels particularly from former customers, in the form of spare parts and replacements. All such demand has to be integrated into the inventory control

system in addition to normal production demands developed through the material requirements planning system.

4.3 Nature of Supplies and Suppliers

The selection of suppliers for any company's inventory requirements is preceded by numerous considerations. These include the reliability of the supplier, supply lead times, quality of the supplier's products, costs and other inter-corporate or even personal relationships. In developing the inventory program, it has been assumed that the essential criteria has been used and the selection made. The necessary information is printed in a vendor file in accordance to pre-allocated vendor codes. The same vendor codes are used by the order processing subroutine for direct access of the relevant information. In order to maximize file space utilization the codes are serially allocated between 001 and 999 inclusive.

4.4 Inventory Annual Cost Considerations

The task of processing inventory usage and replenishment through the inventory control system can be uneconomical for some of the low cost inventory items. The concept of an ABC analysis has been associated with the categorization of inventory with a view to separate those items that demand close attention from those that do not. The following subsections deal briefly with the question of preferred treatment of inventory items.

4.4.1 High Cost Items

The inventory items that represent most of the dollar value are the ones that obviously require most attention. Carrying unnecessarily large inventory quantities means money tied up that could otherwise be invested elsewhere. The literature suggests that these 'vital few' items constitute about 10% of the total inventory items and 70% or more of the total inventory investment[14]. Company considerations, however, should dictate the coverage and extent of control that is required. Such considerations should include the total number of inventory items, cost reduction associated with preferential control and the importance of individual inventory items in the production process. Although low cost items may be loosely controlled by a "bin" system, it is important that these items are available when needed, as will be discussed in the next section.

4.4.2 Low Cost Items

Depending on company requirements, some low cost items may be excluded from the computerized control system without much impact on the overall performance, provided that the supply of these items is assured at all times. However, in order to make maximum use of the report generating capabilities of the system as well as its order processing routines, it may be a worthy undertaking to include some data pertaining to these items in the system data base. This

information can be updated on a periodic basis without having to process the day-to-day transactions on these items. The system can thus, be used in processing orders and other applicable system functions.

4.4.3 Minimum Economical Order Quantities

Some authors have argued that the economic lot size concept is an old concept that is no longer useful on the eve of the now growing use of material requirements planning systems[15]. The argument is based on the observation that inventory cost is rather insensitive to small variations of the lot size and their conviction that with MRP the placement of orders so that supplies arrive just-in-time (the JIT concept) is a reality.

The insensitivity to cost around the economic order quantity(EOQ) as mentioned above, allows managers to make necessary adjustments to the calculated order quantities, noting that some of the figures reached may not be practical. This should not be used as a reason to drop the use of EOQ calculations. With regard to the second reason, one notes (from the remarks made in the introduction to this chapter), that even with MRP, safety stocks may still be necessary as a result of cost reduction in order placement or as a means to handle the unexpected. One should note that in the real business environment things aren't deterministic, therefore one is dealing with forecasts or estimates of demand, supply lead times, production volumes,

scrap levels and inventory replenishment costs. In such an environment, the JIT concept may not offer the expected benefits.

The inventory control system programs (the listing is given in appendix C) contain some subroutines for the evaluation of economic order quantities. These are provided as an aid, not as a solution to the problem. Order quantities may over-ride the EOQ figure. This is in appreciation of the fact that order quantities may be influenced by so many factors, like volume discounts and shipment considerations that are not accommodated in the computer routines. The key point to remember is that the calculated figure will only be as realistic as the parameter estimates used for its evaluation, which should be input and constantly updated by the users.

A description of the different programs in the inventory control system is given in the next section of this chapter.

Procurement Lead Times.

For every inventory item that is held in the computer system, a figure (in days) for the purchase or manufacturing lead time should be stored. This should be based on latest order durations applicable to the main supplier for the particular item. The figure is useful in the estimation of order arrivals and thus future on-hand positions.

4.5 The Inventory Control Programs

The inventory control system is supported by three major programs which were written with a view to provide for the necessary inventory transactions, enquiries, data processing, order processing and the generation of a variety of important reports. The program, written in fortran, was developed on the IBM micro-computer. It is thus intended for use on such machines. The figure below illustrates the breakdown of the computer programs.

INVENTORY IDENTIFICATION AND CONTROL SUBSYSTEM.

PROGRAM: MAIN11	PROGRAM: MAIN22	PROGRAM: MAIN33
(General Transactions Alterations, Sorting Order-processing etc.)	(Regular reports, Parameter evaluation & exception reporting)	(Inventory Identification/ coding and Sorting)
Prog units employed	**Prod units employed**	**Prod units employed.**
CHGREP	DATPRO	ENTRES
DALTER	DATVER	MAIN33
DATVER	DREAD	(PATCOD)
DREAD	ENTRES	RTHEAD
DWRITE	EXCEPT	SORTIT
ENQURY	INTROD	THEAD
ENTRES	MAIN22	
EXCEPT	REGREP	
INTROD	RTHEAD	
MAIN11	THEAD	
NEWDAT		
OPTION		
ORDERS		
ORDSLP		
PROMPT	NB: The different subroutine functions are described in APPENDIX B	
RTHEAD		
SORTIT		
THEAD		
TRANSN		
VARCOD		

Figure 4.1 Illustration of Program Units Used in the Three Inventory Control Programs

To ensure maximum flexibility and/or versatility, the following approaches were employed in the development of the programs.

1. The program was written in the form of many small subroutines performing different specific tasks. This ensures minimum duplication of tasks, if any, by different sections of the program. By 'calling' the specific subroutines whenever these tasks are required, program size can be reduced without reduction in its capability.
2. The program was written in several files, (thirteen in all) and compiled separately before they were 'linked' in the run file. This is very useful in the process of debugging the program or when any additions have to be made. That is due to the need only to re-compile the particular file requiring debugging and/or additions. When this has been accomplished the object files can then be re-linked to obtain the needed run file.
3. Data sharing between different program units (the main program and the subroutines) is achieved through common blocks instead of the more conventional argument type transfer. The use of common blocks reduces memory space requirements since each variable common to a number of program units is only stored in one location in memory instead of the separate locations for each argument necessary with the other approach.
4. Data access from storage is accomplished through "direct

read" as opposed to sequential reading of data. This is very useful for fast access and repetitive reading requirements that are quite cumbersome with sequential reads. Direct access writing, on the other hand, is very useful noting that after any inventory transaction or parameter variation the new status has to be written in the same location where this information was originally stored. This is, thus, achieved without re-writing any other information.

Figures 4.1 and 4.2 show the different program files used for the three inventory control programs. Thirteen different program files are used in all, the size of the source files ranging from 1,354 bytes for source file MSCEE.FOR, to 10,178 bytes for source file MSC11.FOR. The source files together constitute a total of 71,484 bytes. The biggest execution file is the one for the first program which occupies 85,672 bytes. The second and third execution files comprise 65,664 and 54,784 bytes respectively. The combined size of the execution programs for the inventory control system is 206,080 bytes.

Source File Name		Size(bytes)
MSC11	.for	10178
MSC22	.for	6785
MSC33	.for	5281
MSC44	.for	5787
MSC55	.for	2705
MSC66	.for	3735
MSC77	.for	5838
MSC99	.for	3371

Object File Name		
MSC11	.obj	12599
MSC22	.obj	10369
MSC33	.obj	7169
MSC44	.obj	10765
MSC55	.obj	5619
MSC66	.obj	6608
MSC77	.obj	9478
MSC99	.obj	5529

Execution File		
MSC11	.exe	*85632*

Figure 4.2 Program 1: Source, Object and Execution Files

Source File Name	Prog #	Size(bytes)
MSCBB.for	2	4696
MSCDD.for	2	7697
MSCEE.for	2 & 3	4445
MSCFF.for	2	1358
MSC77.for	2	5838
MSCAA.for	3	9608
MSC99.for	3	3371
Object File Name		
MSCBB.obj	2	6917
MSCDD.obj	2	12526
MSCEE.obj	2 & 3	5752
MSCFF.obj	2	2783
MSC77.obj	2	9478
MSCAA.obj	3	13692
MSC99.obj	3	5524
Execution File Names		
MSC22.exe	2	65664
MSC33.exe	3	54784

Figure 4.3 Program 2 & 3: Source, Object and Execution Files

4.5.1 General Transactions and Order Processing

The first of the three programs used for the inventory control system (and the most extensive one), deals mostly with general inventory transactions and order processing applications. As evident in Figure 4.1, this program employs twenty subroutines. The tasks accomplished by each one of these subroutines are described in appendix B of this report.

As soon as the program is accessed, the subroutine ENTRES is first executed. At this point the user inputs relevant user information like name, date and user code. The subroutine DATVER is then executed. This subroutine first ensures that the user code supplied is valid, returning control to the main program if three attempts are abortive, at which point the execution of the program is terminated. Some of the characters in the date are manipulated to provide the necessary format type for system storage purposes. The third subroutine to be executed is the subroutine EXCEPT, which generates an exception report based on parameters evaluated from the information in the data base that require prompt action. These include such conditions as inventory quantity levels that have fallen below a predetermined re-order level (ROLEV) and orders that are overdue.

After the execution of the exception report generating subroutine, the user is prompted to choose any one of the following major subroutines depending on the kind of action

desired.

1. The main transaction subroutine (TRANSN), for changing any inventory records or making any inventory transactions.
2. The enquiry subroutine (ENQURY), provides information pertaining to any individual inventory item stored in the system data files.
3. The order processing subroutine (ORDERS), provides for the placement and receipt of both purchase orders and in-house work orders.
4. The inventory sorting subroutine (SORTIT), is used to generate a listing of inventory items with a specified combination of significant characters in its identification code.

Each one of the above major subroutines employs several of the other smaller subroutines in performing its tasks. The subroutine INTROD and PROMPT are accessed by most of the other subroutines at different stages to provide necessary information to the user as well as to seek relevant inputs to the system. The sequence of executions relating to the first program of the inventory control system can be illustrated according to the flow diagram in Figure 4.4, with only the major sections shown.

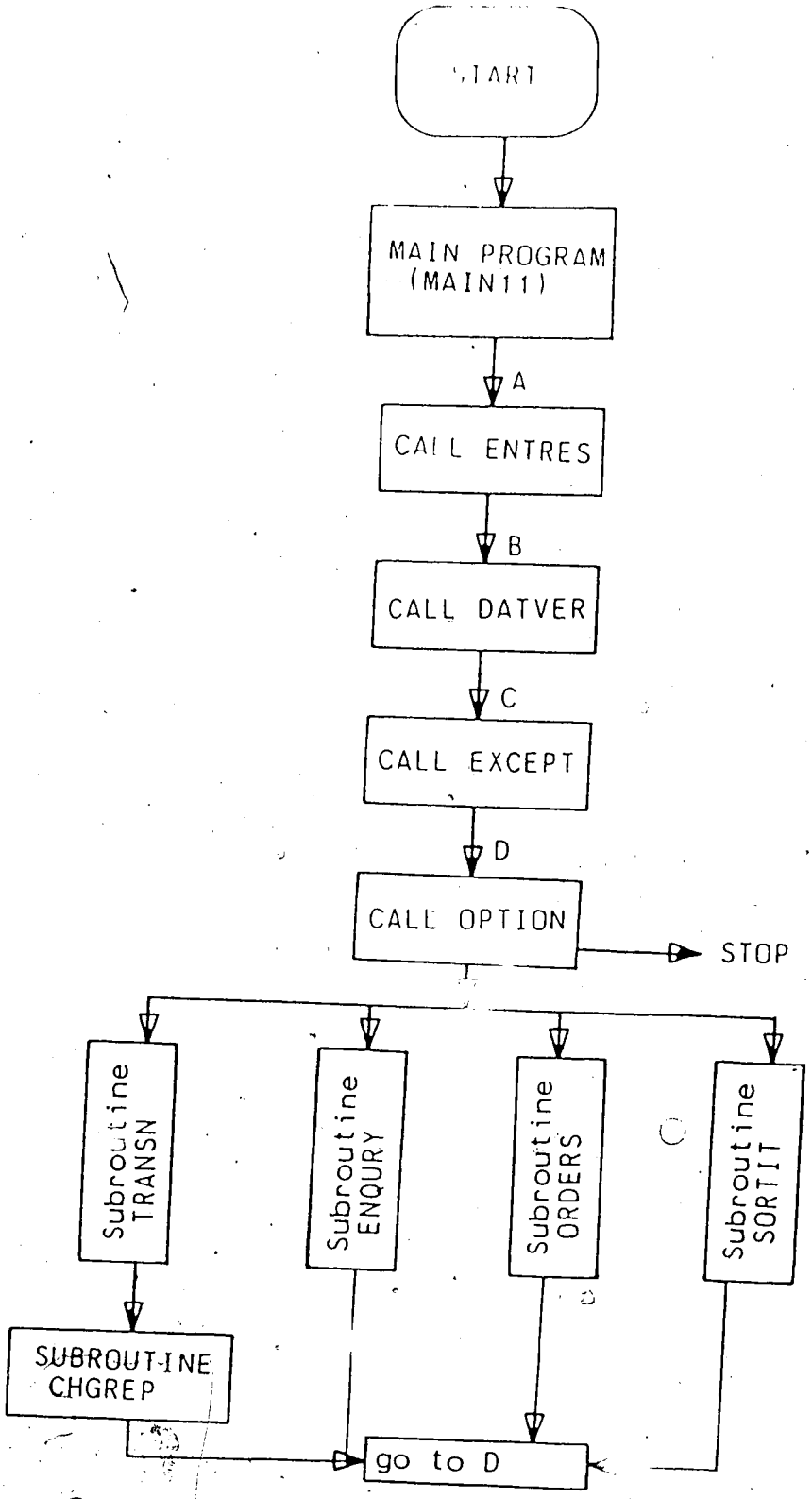


Figure 4.4 Flow Diagram for the First Inventory Control Program

4.5.2 Regular Reports and Parameter evaluation

The second of the inventory control programs deals with the generation of exception reports (as was the case with the first program), the generation of regular reports, and the evaluation of economic order quantities and inventory safety stocks (buffer stock), where applicable. The initial sequence of executions on program entry is similar to the first program, ie. the subroutines ENTRES, DATVER and EXCEPT in that order. After these the user then has the option to pick for execution one of the two major subroutines in this program, regular report generation or parameter evaluation.

If the regular report generating subroutine is opted for, which would normally be done at the end of the week or month, a comprehensive report of the status of each one of the inventory items in the system is generated. This report provides all key data and information pertaining to the inventory items.

The parameter evaluation subroutine handles the economic order quantity (EOQ) and the inventory safety stock problems using two distinct approaches. For the first problem the basic economic order quantity formula is used, for which the incremental cost on inventory is expressed as a function of the total carrying cost and the total procurement (ordering) cost. In order to evaluate this quantity, the inventory carrying cost per unit per year, the inventory annual requirements, as well as the procurement cost per order are required. Normally these figures will be

available as part of the system data for the particular item. The user will however have the option to vary any one of these parameters at the time of each evaluation.

The second approach used in the parameter evaluation subroutine is one of probabilistic demand considerations over the lead time. Given a particular desired customer service level and a known mean lead time demand (in this case lead time assumed constant), the necessary buffer stock (and therefore reorder level) is determined. The evaluation is based on the constraint that the probability of a stock-out may not exceed the predefined service level. If the desired service level was 95%, then this probability must not exceed 5%. In this routine an assumption of a normal demand distribution over the lead time is employed. This is a necessary, and no doubt a reasonable assumption to enable one to treat the probabilistic problem.

The use of this routine in the evaluation of re-order levels must be done with caution. The assumption of a normal distribution of demand over the lead time period does not hold for low volume items. The basic concept of a normal distribution is derived from an assumed large population. Thus, while high volume inventory items would be expected to show a normal frequency distribution of demand around a mean value, the same can not be generalised to include low volume items. At the same time one must be satisfied that no conditions unique to the particular planning period are expected to influence item demand. If any special factors

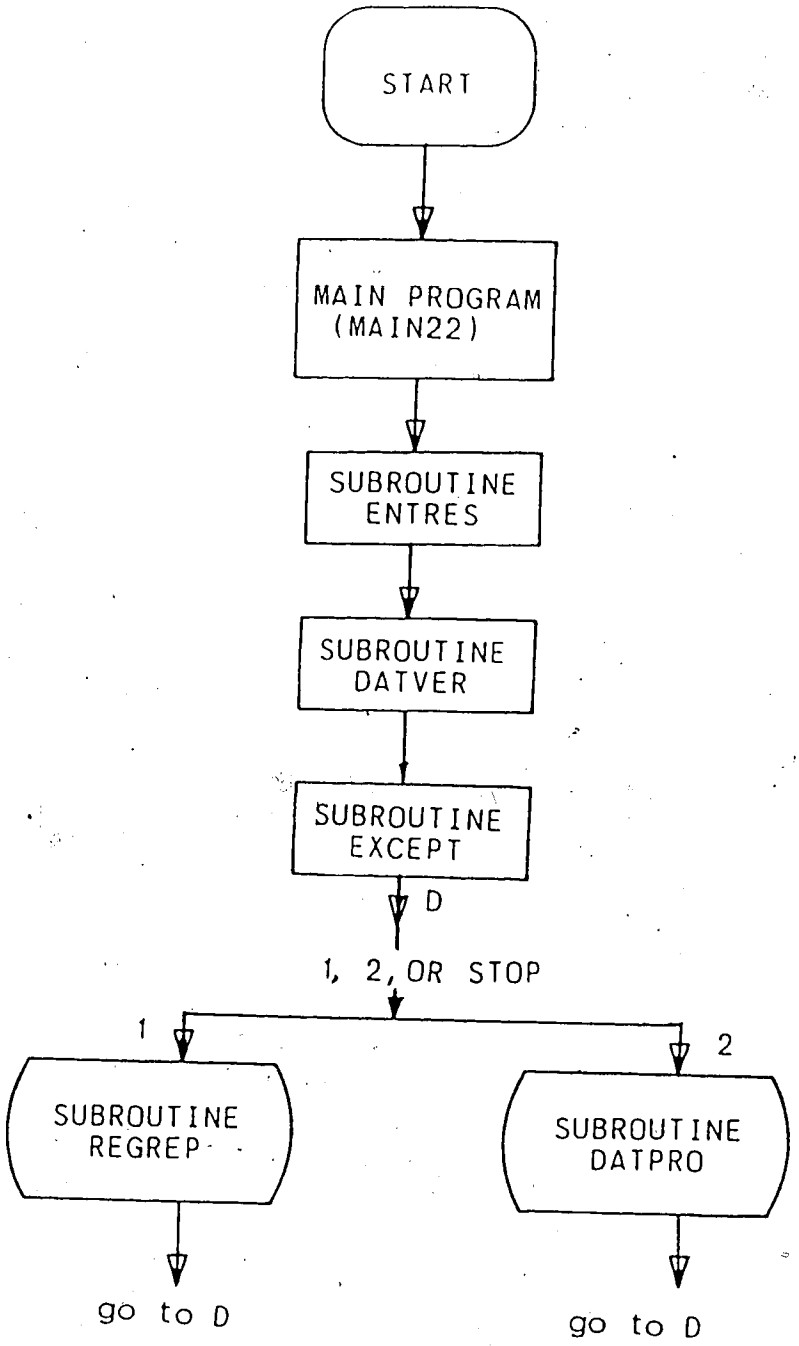


Figure 4.5 Flow Diagram for the Second Inventory Control Program

are expected to influence specific item demand, one should seek to establish and incorporate the anticipated variations in the scheduled inventory requirements.

The service level figure used in this subroutine, as described above, is merely a user-defined level of performance. This figure does not represent the actual level of service accorded customers. By defining, say, a ninety five percent service level for a particular item, the inventory manager is in effect saying that this is the lowest level of service that would be acceptable to the company. The actual level of service to customers would normally be different from this figure, depending on customer order patterns and other demand variations. Customer service level (or service ratio) can be defined as the ratio of customer orders , say in dollar figures, supplied when needed or as scheduled, to the total amount of orders placed.

The two applications described above, are some of the more basic ones in inventory control models. Their simplicity makes them more acceptable in industry because of the small number of parameters that need to be estimated. These models are for most applications more reliable for similar reasons. Since the parameters that go into the evaluation are not deterministic, to be on the safe side one usually keeps the number of such parameters to its minimum.

Figure 4.5 illustrates the sequence of executions followed in the second inventory control program.

4.5.3 Inventory Identification and Sorting

The last of the inventory control programs is the inventory identification and sorting program. The inventory identification procedure was illustrated in Figure 3.5 of the preceding chapter. The same procedure is employed in the development of the inventory identification subroutine.

The sorting subroutine functions as follows. First, the user must decide which item characteristics are to be sorted amongst those built into the item codes by way of the significant characters. Any combination of significant characters can be used for the sorting procedure. Blanks are input for those fields for which sorting is not required. Thus, if all items with the characters "B" and "2" in the first and third positions respectively were to be sorted, the entry that has to be made is "B 2 ", noting that the inventory codes only have four significant characters. The program then searches through all inventory codes in the data base, creating an array of those items with the required combination of significant codes. This array can subsequently be printed on the printer or on the screen at the users descretion. The sorting subroutine, as noted earlier, is also available in the first program.

Figure 4.6 illustrates the execution sequence followed in this inventory control program. The next section will look at some of the outputs from the inventory control system programs.

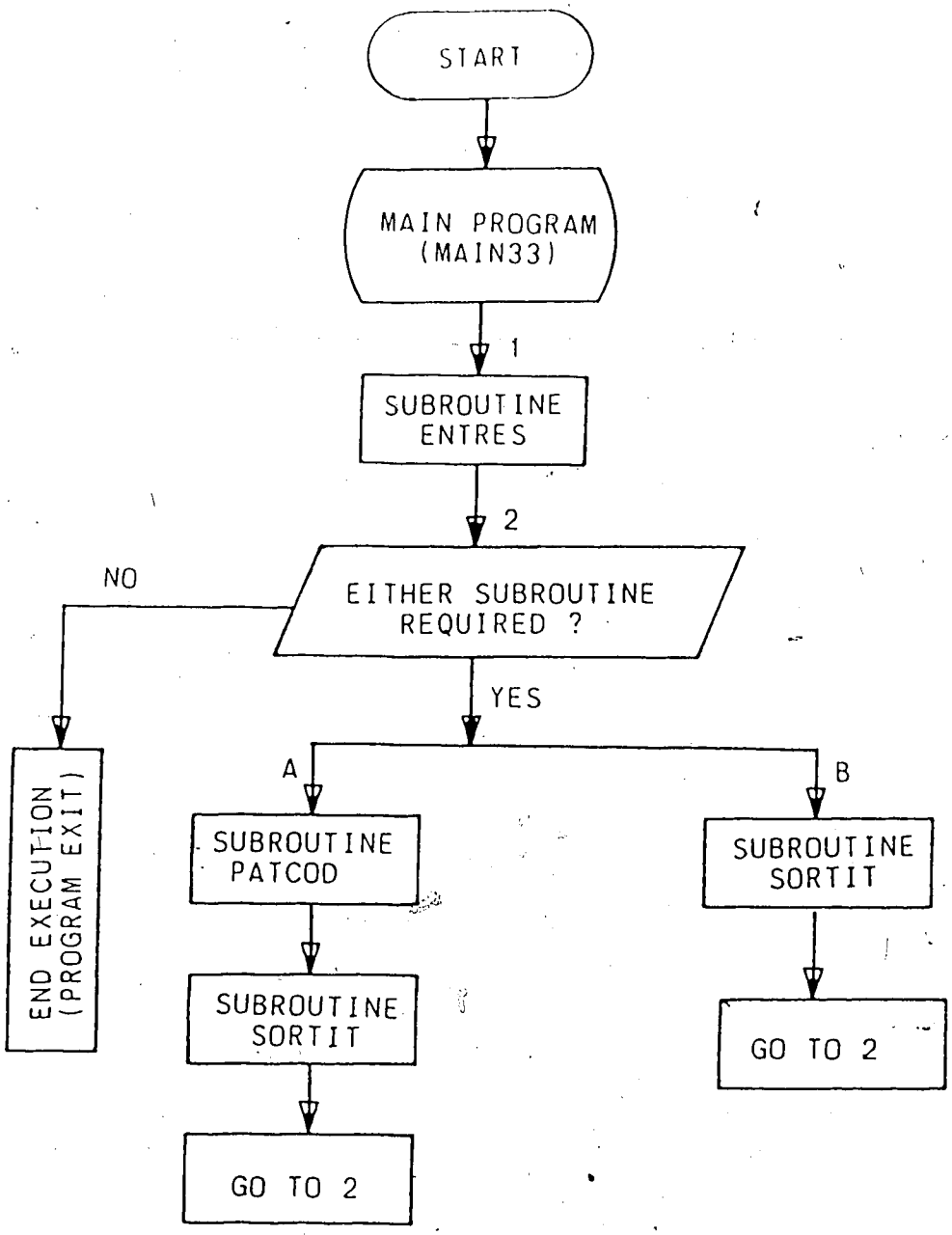


Figure 4.6 Flow Diagram for the Third Inventory Control Program (Inventory Identification and Sorting)

4.6 Outputs from the Inventory Control System Programs

Previous sections have concentrated on the different tasks performed by the three inventory control programs. Each of the three programs was illustrated (by a flow diagram), showing the main routines executed by the program. To provide a clearer picture of the programs' coverage, some outputs from the computer system will be very useful.

In the following pages, some of the system outputs from the inventory control system are presented.

Figure 4.7 illustrates entry to one of the execution programs and also shows an exception report. The execution program being run is the one in the execution file MSC11.EXE, which in this case is contained in drive 'B' of the micro-computer disk drives. It should be noted that the extension 'exe' does not need to be entered during the run command. After executing the subroutines ENTRES and DATVER, the execution proceeds to the generation of the exception report. A code key is printed below the report for those exception codes appearing in the report. The exception report would also be generated if the second program (MSC22.EXE) was executed instead of the first one. The procedure given at the beginning of this section is followed in this figure.

Figure 4.8 shows the format for the regular reports. As explained in the previous section, the regular report would

normally be generated on a periodic basis, say bi-monthly or monthly. This report is a hard copy of the inventory status of all inventory items recorded in the system data base. Most of the information pertaining to the items in stock is output for the company's records.

For each individual inventory item, the part number and item description are first printed, followed by twenty eight other data and information. In view of the volume of these reports, it is advisable only to seek such reports in intervals of one month or so, noting that individual inventory information can be printed by the system whenever required through the enquiry subroutine (ENQURY).

To obtain the regular report, the second inventory control program is executed, according to the procedure earlier outlined. The appropriate option for generating the report is input by the user following the relevant prompt from the system. (See appendix D)

```
A>B:MSC11

THE DEPARTMENT OF MECHANICAL ENGINEERING-- UNIVERSITY OF ALTA.
==============================================================


PLEASE ENTER YOUR INITIALS (8 CHARACTERS)
EM.KOMBE

ENTER TODAY'S DATE (M/D/Y -eg.04/22/83)
07/19/83

ENTER YOUR PERSONAL USER ID. CODE
A9

RECORDED DATES IN DATABASE== 71983 &  719 (m/d/yr)

DO YOU WANT EXCEPTION REPORT PRINTED ? Y/N?
Y

          ************************************************
          **                                            **
          **  ARGO ENGINEERING LIMITED      DATE:(M/D/Y)  **
          **                                    07/19/83 **
          **  THIS IS THE:                              **
          **  ===>> EXCEPTION   REPORTING  SUBROUTINE   **
          **                                            **
          ************************************************
!!!!!!!! MAKE SURE PRINTER IS ON..!!!!!!!
Press <enter> to continue.

0.....HOLD ON ! EXCEPTION REPORT IN PROCESS...


MWM                                                                  MWM
MWM   ARGO ENGINEERING LIMITED.                  Date:(mo/da/yr)     MWM
MWM   =======================                  +++++++++++++++     MWM
MWM   (Specialists in Material Handling Equipment.)    07/19/83     MWM
MWM                                                                  MWM


          ################################################
          #####                                      #####
          #####  **E X C E P T I O N   R E P O R T**  #####
          #####                                      #####
          ################################################


ITEM NUMBER:   CODE1   CODE2   CODE3   CODE4   CODE5   CODE6   CODE7   CODE8
-------------------------------------------------------------------------
    A23B1452           LA
    A14B2969     SL
    A22C1419           LA      LA
    A84A0003     SL    LA      LA
    B12D1001           LA      LA      LA
    B34A0100     SL
    K43B2016     SL    LA      LA
    C01C1301     SL
    K43B2016     SL

  CODE KEY:
-----------
```

Figure 4.7 Program Entry and Exception Report

```
##################################################
#####                                        #####
#####  ** R E G U  L A R   R E P O R T **    #####
#####                                        #####
##################################################
```

PART NUMBER: A25B1452

PART DESCRIPTION: BOLT 2X3X8CM/STLHD-M

ON HAND	UNIT COST	HOL COST	ORD COST	EC.ORD.QTY	RE ORD LE	PROC/LTIME
325.00	50.00	12.00	34.00	1000	200	36
AVER DEM	SERV LEV	INV LEV	M OR BUY	INV CLASS	MEAS UNIT	DEPT COD
10000	.95	23	B	B	EA	D4PRE
LOCN CODE	LOCN NAME	ORD NO 1:	ORD.1 DAT	EXP. ARR	ORD.1 QTY	
F6	STOR1/L6	34305	511	618	560	
ORD # 2	DATE:	EXP.ARR	QUATTY	ORD # 3	DATE:	EXP.ARR	QUATTY
35536	719	825	336	0	0	0	0

PART NUMBER: A14B2969

PART DESCRIPTION: STE.WELD/FRAM-12X48T

ON HAND	UNIT COST	HOL COST	ORD COST	EC.ORD.QTY	RE ORD LE	PROC/LTIME
359.64	50.00	12.00	34.00	1000	800	36
AVER DEM	SERV LEV	INV LEV	M OR BUY	INV CLASS	MEAS UNIT	DEPT COD
10000	.95	23	B	B	EA	M5WDG
LOCN CODE	LOCN NAME	ORD NO 1:	ORD.1 DAT	EXP. ARR	ORD.1 QTY	
F6	STOR3/U8	0	0	0	0	
ORD # 2	DATE:	EXP.ARR	QUATTY	ORD # 3	DATE:	EXP.ARR	QUATTY
0	0	0	0	0	0	0	0

PART NUMBER: A22C1419

PART DESCRIPTION: BOLT 2X3X8CM/STLHD-N

ON HAND	UNIT COST	HOL COST	ORD COST	EC.ORD.QTY	RE ORD LE	PROC/LTIME
359.90	50.00	12.00	34.00	1000	200	36
AVER DEM	SERV LEV	INV LEV	M OR BUY	INV CLASS	MEAS UNIT	DEPT COD
10000	.95	23	B	B	EA	D4PRE
LOCN CODE	LOCN NAME	ORD NO 1:	ORD.1 DAT	EXP. ARR	ORD.1 QTY	
F6	STOR1/L6	33333	424	430	55555	
ORD # 2	DATE:	EXP.ARR	QUATTY	ORD # 3	DATE:	EXP.ARR	QUATTY
32255	509	610	460	0	0	0	0

Figure 4.8 Part of Regular Report Showing Output Format and Detail Covered

Figure 4.9 illustrates some output from a sorting application of the inventory control system. In this particular sort, all item codes that bear the fourth character "B" are printed by the system. Print options are provided in case the user wants to narrow down the sorting. If for instance, one hundred items are in this category, the user may want to go back and make a more specific sort without printing the hundred item specifications. In this case one may want to sort for items with third and fourth characters "2" and "B" respectively. Any combination of significant characters (first, second, third and fourth ID. characters) can be sorted by the system.

Sorting can be done from either program MSC11.EXE or program MSC11.EXE. In both cases, after loading the execution program, the appropriate option should be given to indicate that sorting is required. (See appendix D)

Figure 4.10 shows some of the stages followed in an order processing routine. The output shown here, like in many of the other figures, includes some of the output that is actually directed to the user (through the screen) and some of the input from the user's keyboard. The information given in the vendor slip serves as a check that information has been input to the system correctly, while at the same time serving as a valid order slip to a work centre or supplier as the case may be.

```
THIS PROGRAM HANDLES THE FIRST FOUR CHARACTERS OF THE ITEM CODES.
A SEARCH IS MADE FOR ITEMS/PARTS WITH THE SPECIFIED ID. CODES
IN THE GIVEN POSITIONS. YOU WILL BE TOLD THE TOTAL NUMBER
OF ITEMS IN QUESTION AND ASKED WHETHER THEY SHOULD BE PRINTED
ON THE PRINTER OR ON THE SCREEN

**INPUT THE REQUIRED CHARACTERS--LEAVING BLANKS ELSEWHERE .. eg. D_2_
   B

          TOTAL APPLICABLE #. OF ITEMS=   8
         DO YOU WANT THEM ON PRINTER?
            ENTER Y FOR YES OR
            ENTER N FOR SCREEN O/P OR
            ENTER A BLANK FOR NO OUTPUT AT ALL..
Y


      ##################################################
      #####                                        #####
      #####     MATERIAL  PARTNO  SORT  PRINTOUT   #####
      #####                                        #####
      ##################################################


 ---------------------------------
 SER.#     ITEM ID. NO.     DESCRIPTION
 -----     ------------     -----------
  1        A25B1452
 ---------------------------------
  2        A14B2969
 ---------------------------------
  3        K43B2016
 ---------------------------------
  4        C17B0206
 ---------------------------------
  5        D42B2119
 ---------------------------------
  6        R12B0206
 ---------------------------------
  7        R49B1462
 ---------------------------------
  8        X63B1820
 ---------------------------------
  DO YOU WANT ANOTHER SORTING ? ..Y OR N ?
```

Figure 4.9 Output from Sorting Routine for Invent Items with Fourth ID. Character "B".

```
THE FOLLOWING ORDERS ARE OUTSTANDING (0==>NONE)
---------------------------------- 34305      0      0

ENTER THE ORDER NUMBER FOR THE NEW ORDER [I5]
35536

ENTER THE QUANTITY-[I5] OF THE ORDER # 35536
336

THE PROCUREMENT LEAD TIME ON THIS ITEM IS  36 days.
ENTER THE EXPECTED RECEIPT DATE__I4 [mo/dy]
0825
```

```
##################################################
#####                                        #####
#####  M A T E R I A L    O R D E R   SLIP   #####
#####                                        #####
##################################################
```

```
T O :
-----

ENTER VENDOR/SUPPLIER CODE FOR THE ITEM NO. A25B1452
2

      THE ALBERTA GENERAL
       HARDWARE DEPT.STORE
        EDMONTON, ALBERTA.
       T6K 2M6.
       ---------

ORDER NUMBER          OF DATE[mo/dy]          QUANTITY
------------          --------------          --------

     35536                  719                336   ## EA ##

MATERIAL DESCRIPTION:
---------------------
                    BOLT 2X3X8CM/STLHD-M

                                                by: EM.KOMBE
                                                    --------
                                        for Argo Engineering Limited.
```

Figure 4.10 Illustration of Order Processing Procedure and Material Order Slip for Inventory Item

The order processing routine is one of the options available in the first execution program. To access the subroutine, this option is given after loading the first program, following the option prompt as shown in appendix D. Additional discussion on order processing is contained in chapter five of the report.

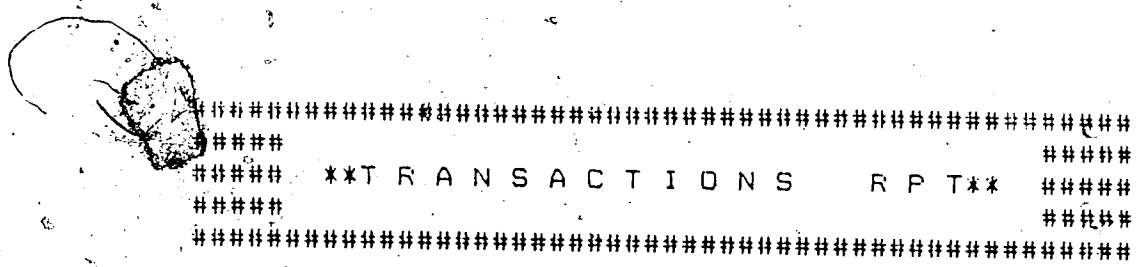
Figure 4.11 shows a transaction report. This is a report that may be printed at the end of inventory input/output transactions at the option of the user. The report shows all the inventory items for which transactions were made from the beginning of the particular program run. Only some of the key information about the inventory items is shown in this report. This information includes the transaction number, the item number (identification), the new inventory on-hand position, the re-order level(ROLEV) and the outstanding order quantities for orders one through order three.

The transaction report is generated from the main transactions subroutine. This major subroutine, is part of the first execution program (see appendix D). At the end of a series of transactions, the system queries the user on whether the report should be printed. The report is printed if the user so indicates. To enter the transactions subroutine, one loads the first program and indicates the appropriate option to the system, as illustrated in appendix D.

Figure 4.12 depicts the data storage format in the three main inventory files - DATA1.INV, DATA2.INV and DATA3.INV, of the system data base. The data files are contained in a diskette labelled 'DATA'. The variables for which information is stored in these files and the data formats are as follows below. Storage of new item information in the data files is accomplished through the transactions subroutine. To do this, enter this subroutine following the procedure given, and enter the new item identification code when prompted by the system. The system will seek confirmation that this is a new item, after which the subroutine "NEWDAT" is used to enter and store the data pertaining to this item.

To make any transaction or change any inventory information, enter the inventory ID. and pick one of the options listed by the system. *(the description of the given variables is in appendix A of the report)*

1. **File: DATA1.INV**
   a. Variables: PARTNO, NAME, UNIT, MOB, CLASS, DEPCOD, DATE1, DATE2, CODLOC, LOCNAM
   b. Format: (A8, 1X, A20, 1X, A4, 2(1X,A1), 1X, A5, 2(1X,I6), 1X, A2, 1X, A8)
2. **File: DATA2.INV**
   a. Variables: SOHAND, HOLCOS, COSTOD, COST1, SEVLEV, EOQTY, ROLEV, PLDTIM, AVDEM, INVLEV
   b. Format: (F8.2, 2(1X,F6.2), 1X, F8.2, 1X, F4.2,



TRANSACTIONS RPT

TOTAL TRANSACTIONS = 7

TRANS.#	ITEM.NUMBER:	NEW O/HAND	ROLEV	OUTSTANDING ORDER Q/TITIES		
1	A25B1452	325.00	200	560	0	0
2	A14B2969	360.00	800	0	0	0
3	A84A0003	296.00	800	45455	55550	0
4	B12D1001	266.00	200	45455	55550	55660
5	B34A0100	325.00	800	0	0	0
6	C01C1301	245.00	800	0	0	0
7	A22C1419	359.90	200	55555	460	0

********** END **********

DO YOU LIKE T@ RE-ENTER THE MAIN PROGRAM?? Y/N
*N

THANK YOU [[EM.KOMBE]] FOR USING THIS PROGRAM.

Figure 4.11 Transaction Report - Indicating the Changed Inventory Status

```
A>TYPE DATA1.INV
A25B1452 BOLT 2X3X8CM/STLHD-M   EA   B B D4PRE   42183       0 F6 STOR1/L6==
A14B2969 STE.WELD/FRAM-12X48T   EA   B B M5WDG   42183       0 F6 STOR3/U8==
A22C1419 BOLT 2X3X8CM/STLHD-N   EA   B B D4PRE   42183       0 F6 STOR1/L6==
A84A0003 BRNG-HOSNG/25D/DB.SC   EA   B B MACHN   40483       0 F2 GENSUPLS==
B12D1001 BOLT 2X3X8CM/STLHD-N   EA   B B D4PRE   42183       0 F6 STOR1/L6==
B34A0100 BOLT 2X3X8CM/STLHD-N   EA   B B D4PRE   42183       0 F6 STOR1/L6==
B74A0040 BOLT 2X3X8CM/STLHD-N   EA   B B D4PRE   42183       0 F6 STOR1/L6==
K43B2016 WASHER BR/4x.5x8-STD EACH B B 3MACH   61083       0 G4 STR6/CTR..
C01C1301 BOLT 2X3X8CM/STLHD-N   EA   B B D4PRE   42183       0 F6 STOR1/L6==
K43B2016 WASHER BR/4x.5x8-STD EACH B B 3MACH   61083       0 G4 STR6/CTR..


A>TYPE DATA2.INV
  325.00  12.00  34.00    50.00  .95  1000   200   36    10000  23=======
  359.64  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  359.90  12.00  34.00    50.00  .95  1000   200   36    10000  23=======
  296.00  12.00  34.00    50.00  .95  1000   300   36    10000  23=======
  266.00  12.00  34.00    50.00  .95  1000   200   36    10000  23=======
  325.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  450.00  12.00  34.00    50.00  .95  1000   300   36    10000  23=======
  245.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  450.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  450.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  450.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======
  450.00  12.00  34.00    50.00  .95  1000   800   36    10000  23=======


A>TYPE DATA3.INV
34305  511  618   560 35536  719  825   336     0    0    0     0========
    0    0    0     0     0    0    0     0     0    0    0     0========
33333  424  430 55555 32255  509  610   460     0    0    0     0========
33335  422  429 45455 33337  420  430 55550     0    0    0     0========
33335  422  429 45455 33337  420  430 55550 33445  418  428 55660========
    0    0    0     0     0    0    0     0     0    0    0     0========
    0    0    0     0     0    0    0     0     0    0    0     0========
33335  422  429 45455 33337  420  430 55550 35300  615  722   360========
    0    0    0     0     0    0    0     0     0    0    0     0========
    0    0    0     0     0    0    0     0     0    0    0     0========
    0    0    0     0     0    0    0     0     0    0    0     0========
    0    0    0     0     0    0    0     0     0    0    0     0========
```

Figure 4.12 Data Storage Format for Three Main Datafiles in the Inventory Data Base

2(1X,I5), 1X, I4, 1X, I8, 1X, I3)

3. **File: DATA3.INV**

   a. Variables: ORDRNO(1), ORDRDT(1), EXPTAV(1), ORDRQT(1), ORDRNO(2), ORDRDT(2), EXPTAV(2), ORDRQT(2), ORDRNO(3), ORDRDT(3), EXPTAV(3), ORDRQT(3)

   b. Format: (I5, 2(1X,I4), 2(1X,I5), 2(1X,I4), 2(1X,I5), 2(1X,I4), 1X, I5)

The formats for the other files used for the different program sections are described in the corresponding sections of the report.

In figure 4.13 the enquiry routine application is portrayed. This routine is entered by loading the first program and indicating the enquiry option. Appendix D shows the different execution options given by the system when this program is loaded. When the subroutine is entered, the user is prompted to indicate whether detailed information or only partial information is required regarding the particular inventory item. If detailed information is required, the system prints all relevant information regarding this inventory item.

If, on the other hand, only partial information is desired, as is the case in this figure, the system prints some of the key information and prompts the user to indicate if more information is required. If one did not get the

```
ENTER ONE [1] IF YOU WANT DETAIL INFORMATION ON THE PART NO. A84A0003
OR ENTER ANY OTHER DIGIT IF ONLY PARTIAL INFORMATION IS REQUIRED.
3

CONTROL PrtSc IF PRINTER OUTPUT IS NEEDED !
Press <enter> to continue.

OPART NUMBER:    A84A0003
                 ----------

PART DESCRIPTION:  BRNG-HOSNG/25D/DB.SC

 ON  HAND  UNIT COST HOL COST  ORD COST EC.ORD.QTY RE ORD LE PROC/LTIME
 ********  ********  ********  ********* ********** ********* **********
    296.00     50.00     12.00      34.00       1000       800         36

DO YOU NEED FURTHER INFORMATION ON ITEM # A84A0003 ?
Y


  AVER DEM  SERV LEV   INV LEV  M OR BUY INV CLASS MEAS UNIT  DEPT COD
  ********  ********   ********  ********* ********** *********  *********
     10000       .95         23         B          B        EA      MACHN

DO YOU NEED FURTHER INFORMATION ON ITEM # A84A0003 ?
Y


 LOCN CODE LOCN NAME ORD NO 1: ORD.1 DAT EXP. ARR  ORD.1 QTY
 ********** ********** ********* ********** ********  ********
       F2   GENSUPLS    33335        422       429       1553

DO YOU NEED FURTHER INFORMATION ON ITEM # A84A0003 ?
N


IS INFORMATION NEEDED FOR (AN)OTHER ITEM(S) ?
```

Figure 4.13 Illustration of the Enquiry Reporting Process for Inventory Item A84A0003

information that was required, typing in a "Y" for YES will make the system print some more information. As soon as one gets whatever data was been sought, typing in an "N" for NO following the prompt will cause the execution to jump back to the beginning of the subroutine, so a different inventory item can be handled if needed.

The average demand (AVER DEM) on this report is an annual average as described in appendix A.

NB: Only a representative sample of system applications is given by way of these figures. The descriptions given in the different program sections attempted to show the extensive coverage of the inventory control programs. A study of the program listing given in appendix C will also serve to provide a more detailed picture of the depth offered by the different program applications.

Data Storage Limitations

As mentioned earlier, the system data files reside on one diskette. The question then, is the number of inventory items for which information can be stored on the diskette. One item takes about two hundred and twenty bytes of storage on the diskette. Since the diskette has a storage capacity of a hundred and sixty thousand bytes, it can provide storage space for about two thousand, two hundred inventory items. More than one diskette can be used for data storage

if necessary, the requirement being that each diskette should have the same file names, and a clear distinction between the inventory data on each diskette must be made. The second requirement is necessary to ensure that the right data diskette is loaded before attempts are made to make transactions and/or any changes to inventory information.

The use of multiple data diskettes, though feasible, can be a tedious and sometimes confusing task. The discussion given in chapter six with regard to possible system expansion, should provide an answer to this limitation. Otherwise, the use of double sided diskettes, which are now available, will double the above capacity.

5. SCHEDULING AND CONTROL OF PRODUCTION

5.1 Introduction

Much of the management decision making process with regard to setting production plans, capacity planning and master scheduling was included in the thesis by B. Lim.[10] Performance forecasting, capacity planning, and master scheduling aided by a material requirements planning system provides for the necessary material, tools and production volume figures. This however, does not eliminate the need for day-to-day planning and record keeping which is essential to achieve the above goals. It is at the shop floor that the plans and schedules have to be implemented. The following sections of this chapter, briefly discuss some of the necessary controls at the shop floor for the full benefits of a computer-aided production control system to be achieved. The discussion centres on the record keeping and information-flow aspects of shop floor performance.

5.2 Purpose

An efficient shop control system is expected, amongst other things, to accomplish the following tasks.[6]

1. Keep track of the work as it progresses through the shop.
2. Update the open order records and report progress.
3. Update the master schedule.
4. Indicate if work-stations and/or personnel are

available.

5. Check material availability before dispatching work to the plant.
6. Change order quantities to compensate for unexpected scrap.
7. Assign work to the operators and work stations.
8. Check and maintain work- efficiency and other essential statistics.

Approaches taken by different production shops in dealing with the problem of shop control vary. They range from mere inspection and follow-up by a supervisor to fully automated, on line, shop-status reporting and control systems. Which one of the many variations is economical and effective for a particular shop situation depends on individual production characteristics.

The following sections will concentrate on those aspects of control like purchase orders, work orders, storage, loading and sequencing which ensure the reliability and good performance of the total computer-aided system.

5.3 Purchase and Work Orders

The reliability of the inventory control system as well as the other production control aspects require that action be taken as soon as the need is noted, and that records be updated immediately following changes. The prompt placement of orders(purchase as well as internal work orders) and recording the arrival or completion of these orders is an

important requirement for achieving a high degree of performance.

5.3.1 Placement of Purchase/Work Orders

The inventory control system program contains a major subroutine for order processing applications. A system file "VENDSUP.INV" contains the names and addresses of suppliers of different materials as well as the descriptions of the different work stations within the company where work orders may be sent.

The format for the above entries is as shown in figure 5.1.

The name of the supplier and the address (or the name of a work station and other details) are all printed on one line in the file for each different entry.

Example:

(A): **THE ALBERTA GENERAL**

(B): **HARDWARE DEPT. STORE**

(C): **EDMONTON, ALBERTA**

(D): **T6K 2M6**

Lines A, B, C, and D above appear in the file "VENDSUP.INV" on one line with the entries made in the same order. The first line (A) starts in column 1, the second in column 21, the third in column 41 and the last line (D) starts in column 61 on the same entry line in the file as Figure 5.1 illustrates.

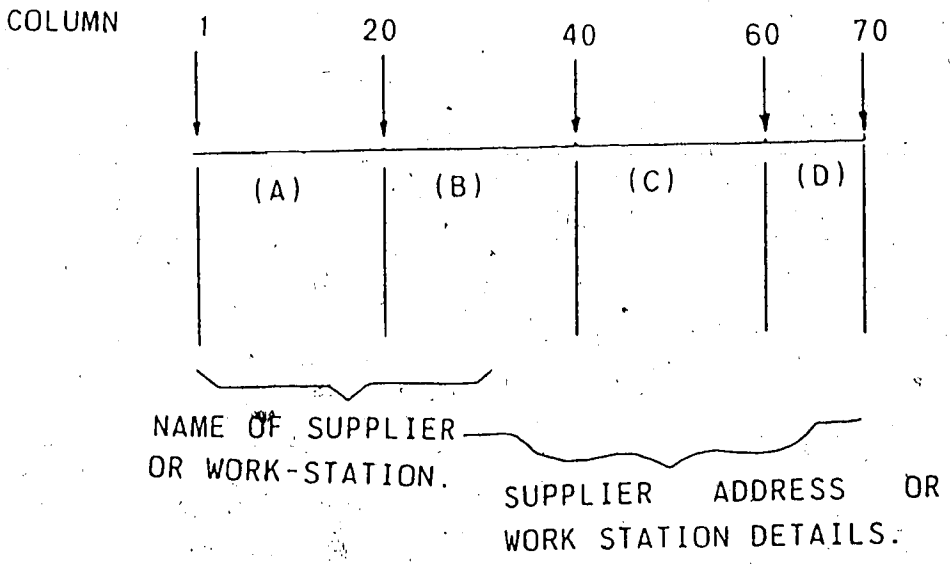


Figure 5.1 Entry Format for Supplier/Work Static Information in the File VENDSUP.INV

The code number for the above supplier is "001", which signifies that this is the first entry in the file. Every other entry, for an outside supplier or an internal work station should have a corresponding code number according to its position in the file. The code assignment procedure is crucial for the proper functioning of this routine. Since direct access reading by the computer is done on this file, when a particular supplier/ work centre code (say code "N") is input, the system goes to the file and reads the information in the "Nth" position of this (direct access) file. When the order slip is printed, its destination (to supplier or work station) appears in the format as shown above.

The system allows for three different orders of any one item at the same time. These orders (order number one, two and three), may be placed to different suppliers and/or work stations. When using the order processing routine, the user is prompted for the supplier (or work station) code. This is the code number for the supplier in the case of a purchase order, or the code number of the work station if it is an internal work order. The code is equal to the serial position of the particular supplier or work station in the file ''VENDSUP.INV' described above. It can be any number between 1 and 999 inclusive. An order slip is printed by the system as soon as the order placement is made, provided a valid code for the supplier or work station (the destination for the order) is supplied by the user. An example of such an order slip is illustrated in Figure 4.10 of chapter four.

An example of the order processing procedure and a work order slip is given below. To place work orders (or purchase orders), enter the option given for the order processing subroutine after loading the first program. The order slip should always be checked to ensure that the correct supplier or work station code was entered. If a code error is made, the order can be cancelled (by indicating a receipt) and a new one placed instead.

```
 ENTER ONE[1] FOR PLACING AN ORDER, OR TWO[2] FOR AN ORDER RECEIPT

 ENTER ANY OTHER DIGIT TO EXIT !
1

NOTE: NO ORDER FOR A NEW INV. ITEM TAKEN !
Press <cr> to continue.

 PLEASE ENTER THE INVENTORY IDENTIFICn #(PARTNO)-[A8]
K43B2016


INVENTORY RECORD NUMBER=     8
------------------------------

THE FOLLOWING ORDERS ARE OUTSTANDING (0==>NONE)
------------------------------------ 33335  33337      0




ENTER THE ORDER NUMBER FOR THE NEW ORDER [I5]
35300


ENTER THE QUANTITY-[I5] OF THE ORDER # 35300


ENTER THE EXPECTED RECEIPT DATE__I4 [mo/dy]
0722


ENTER VENDOR/SUPPLIER CODE FOR THE ITEM NO. K43B2016
  4


MORE ORDERS TO BE PLACED/RECEIVED ? Y/N ?
N


DO YOU LIKE TO RE-ENTER THE MAIN PROGRAM ? Y/N
N
```

****

*** W O R K - O R D E R S L I P ***

****

T O :

WORK-STATION # M/384
MACHINE SHOP DEPT.
(AUTO SPNDL LATHE)

ORDER NUMBER	OF DATE[mo/dy]	QUANTITY	
35300	615	360	##EACH##

MATERIAL DESCRIPTION:

WASHER BR/4x.5x8-STD

by: EM.KOMBE

for Argo Engineering Limited.

5.3.2 Receipt of Purchase/Work Orders

It is just as important to record the receipt of an order as it is to place the order at the right time. If this is not done promptly, outstanding order information in the data base becomes incorrect as well as misleading. When a purchase order is received or an internal work order completed, two transactions should be made in the inventory control system to update system records.

1. The particular order should be cancelled through the order processing (receipt) subroutine of the system. This removes that outstanding order from records as well as rearranges the remaining orders for that particular

inventory if such a rearrangement is necessary. Thus, if there were three outstanding orders on the item and the first one was received, orders two and three would become orders one and two respectively, with no order number three.

2. The main transaction subroutine should be employed to record the increase in stock on hand (SOHAND) resulting from the new receipts. This is vital to ensure that unnecessary orders are not placed due to a failure to update inventory status.

While the receipt of orders is a straight forward matter, the placement of an order is preceded by a number of considerations to ensure that the order placement (also called order release) is made at the most appropriate time. The placement of a new work order for instance, should not be made if there are already similar orders in process and available capacity can not handle the other order at the same time. The new order may have to be delayed to avoid excessive work-in-process in the shop. If necessary, existing orders may be expedited. These and other considerations will be discussed in the following sections.

5.4 Loading, Sequencing and Order Rescheduling

While master scheduling deals with the long term production requirements of the company, shop scheduling handles short term production schedules for the shop floor,

say for the period of one day or a few days. Information from demand forecasts and master schedules is translated into workable shop schedules. Figure 5.2 illustrates in a flow chart the process of developing such schedules.

It will be noted that customer orders are not necessarily used to schedule new work orders. Sometimes customer orders may be met straight from existing inventory, in which case such orders do not cause any scheduling action. The handling of customer orders will be dealt with in section 5.5 of this chapter.

The job priority scheduling procedures discussed in the next section are not built into the computer system. The discussion is intended to provide some tools for the decision making process in this area. Should the incorporation of such an application be desired, a subroutine can be written to handle the calculations involved in evaluating the priority functions and other parameters.

5.4.1 Job Priority Scheduling

It is not always an easy task to make a decision on the job that should be loaded on a machine when it becomes available. When a number of jobs are waiting for the same machine (that is in a queue), some priority rules must be employed in order to ensure that the more urgent jobs are done first. These rules will provide some consistent scheduling criteria.

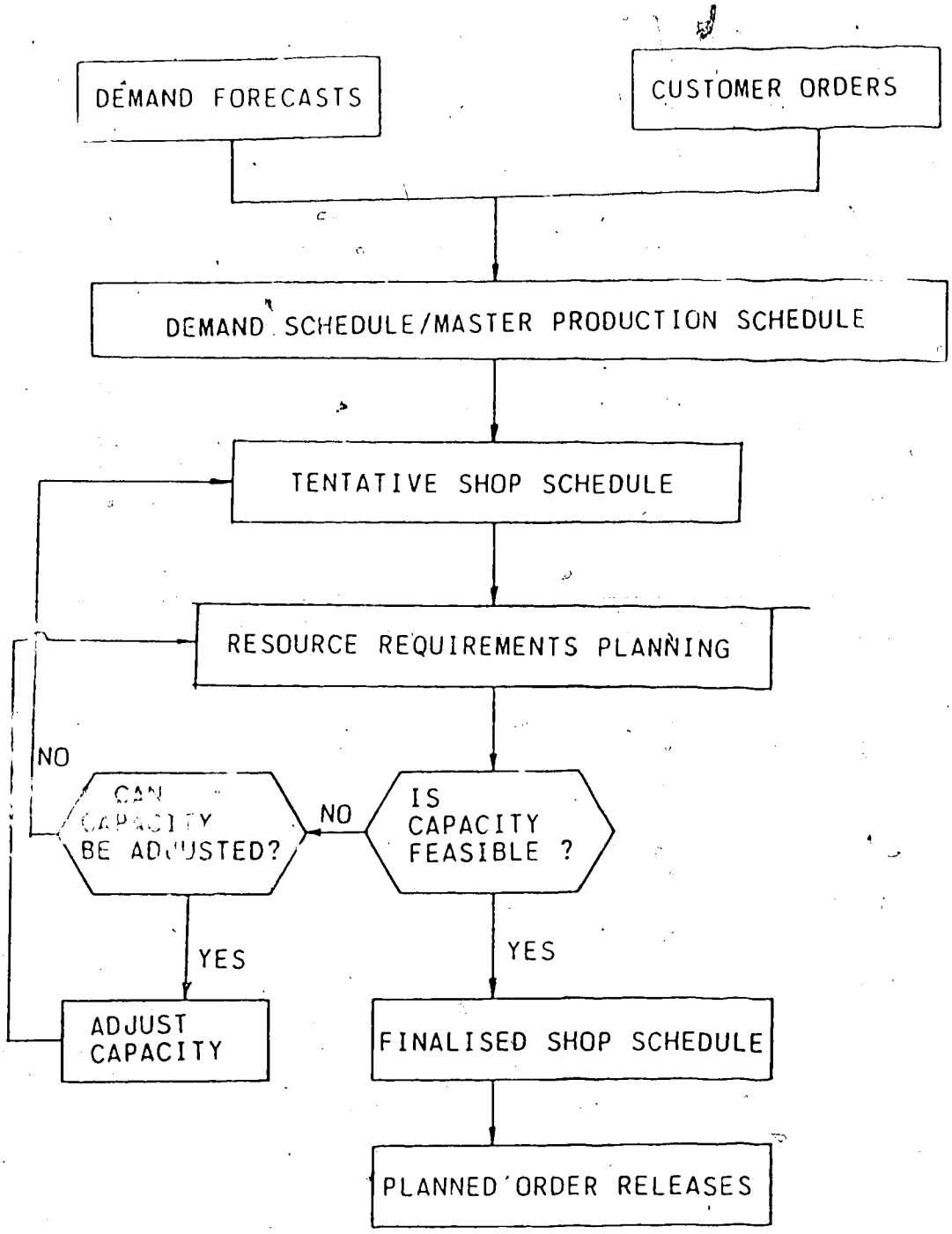


Figure 5.2 Shop Scheduling Process from the Production Master Schedule

A number of priority rules are illustrated in Figure 5.3, with the decision rules for picking the next job.

One frequently used priority rule (the second rule in Figure 5.3) is of the form,

$$\text{Priority Value} = \frac{(\text{Job Slack Time})}{(\text{No. of Oper. Remaining})},$$

$$= \frac{\begin{array}{l}(\text{Due Date} - \text{Present Time} \\ \quad - \text{Remaining Machine} \\ \quad \text{Involvement Time})\end{array}}{(\text{No. of Operations Remaining})}.$$

Rule	Value of Priority Function	Use
Shortest processing time(SPT)	Processing time of the operation for which job is waiting	Load job with shortest processing time first
Slack per remaining operation (S/OPN)	Ratio of remaining job slack time to the number of operations still to be performed	Load job with lowest slack per operation first
Due date of job (DDATE)	Due date for finish of the job	Load job with earliest due date first

NOTE: The 'slack' for a job is the difference between the total time remaining to the job's "DUE DATE" and the total machine operation time required to complete the job.

The "slack per operation" is the slack, as defined above, divided by the number of remaining operations.

Figure 5.3 Priority Scheduling Rules

In picking the priority rule to be used in the shop, first the amount of information required to compute the priority function must be checked against readily available information. If part of the essential information for evaluating the priority function using one of the priority rules, can not be reliably estimated, it is not advisable to use this particular priority scheduling rule. The second consideration is the individual characteristics of the different priority functions. For instance, the second approach, ie. the "slack per operation" is dynamic, which means that priority values are constantly changing with the passage of time as illustrated in Figure 5.4. Depending on the shop requirements, in some instances static measures of the priority function may be preferred to this one.

One notes in the illustration in Figure 5.4 that the job 'M' has higher priority at times 90 and 100, whereas at times 110 and 120 it has lower priority to job 'N'. The times 90, 100 etc. given in Figure 5.4 are relative to the "DUE DATE", which is '200' for both jobs.

The information required for evaluating the priority function would normally be contained in a route card for this job. The priority value can either be pre-evaluated when the number of jobs in a queue is expected to be high, or else, if only a couple of jobs are expected to queue at a work station, the evaluation can be performed by the operator prior to the execution of the next job. The job with the highest priority (lowest priority value function

JOB	DUE DATE	MACHINE TIME	NO. OF OPERATIONS TO BE DONE	PRIORITY FUNCTION WHEN TIME NOW IS:			
				90	100	110	120
M	200	60	10	5.0	4.0	3.0	2.0
N	200	80	4	7.5	5.0	2.5	0

Figure 5.4 Dynamic Nature of the "Slack Per Operation" Priority Rule

since a value of '1' implies higher priority than a value of '2'), is the one that is loaded next as the facility becomes available for a new job.

5.4.2 Rescheduling of Purchase and Internal Work Orders

Rescheduling is the term applied to changing the required due date on a released replenishment order for either purchased or manufactured material. The recommended change can expedite, delay or cancel the order. There are many different situations and/or problems that can lead to a need to reschedule an order. Every company would like as much as possible to do without having to reschedule. Nevertheless, rescheduling is a reality every company has to live with, while trying to keep it to its very minimum.

The following discussion is directed at ways by which companies like Argo Engineering Ltd. can minimise order rescheduling and how reschedules should be handled. The main causes of rescheduling are discussed with the possible remedies.

Amongst the possible causes for rescheduling are the following:

1. changes in the master production schedule,
2. vendor or plant problems, like breakdowns,
3. scrap/spoilage,
4. lot size changes,
5. lead time changes,
6. safety stock changes,

7. changes in engineering design,
8. record errors, and
9. unplanned transactions.

There is no magical solution in dealing with any one of the above causes of rescheduling. Most of them call for a tight control of the overall production process and the maintenance of an information system with high data integrity.

Record errors for instance, can be drastically reduced by ensuring that every transaction is promptly documented, allowing for no side-stepping of the record keeping system. A poor record system not only causes frequent rescheduling, but can easily paralyse the whole system since it inevitably leads to a situation where no one trusts the system. Scrap allowances should be included in the production quantities and any noted variations promptly documented while appropriate action is taken to correct the situation. While it may not be possible to get rid of unplanned transactions completely, these transactions should only be authorised after a thorough study of the consequences. One should also find the best way to handle these transactions, if possible, without having to reschedule other orders.

Changes in system parameters such as replenishment lot sizes for individual items, lead times and safety stock levels can cause excessive schedule changes if allowed to take effect too frequently. Sometimes some of these changes can be delayed without causing much harm to the system,

noting the non-deterministic nature of these variables. Company control personnel should determine the rate and timing of any such alterations to ensure that too many reschedules are not made, particularly when these changes would affect many lower levels in the production process.

Changes in engineering design should be saved for some period of time to be implemented together at some specific time, during which period the applicable schedules will already have incorporated these changes. This approach, while not always feasible, can provide a smooth and effective transition from one design change to the next.

Schedule changes that are a result of problems on the part of the vendor and/or supplier, or a result of unexpected production restrictions in the plant, can not be easily predicted. All one can strive to achieve is to choose the most reliable vendors (according to available criteria!) and ensure appropriate maintenance of plant machinery and equipment. Having done this one can only hope that when these problems arise they will be fixed in the shortest possible time so that the necessary scheduling effort is within the company's capability.

Master production schedule changes constitute an inevitable part of company operations. Thus, one can always expect some level of rescheduling as a result of these changes. The master production schedule is obtained from projections (forecasts) of product demand which in part may contain outstanding customer orders. When either the

forecasts or some of the order quantities and/or delivery times change, it is imperative that the master schedules change. This in turn necessitates rescheduling of shop and/or purchase orders. In the process of rescheduling it is important that an effort be made to ensure that this is done in the most effective and economical manner.

5.5 Processing of Customer Orders

The placement of customer orders is an important focal point of company activity. Even when demand is forecast and production initiated it is always in anticipation of future customer orders. Companies like Argo Engineering Ltd. depend on making the final product for specific customer requirements. While many components may be held in stock, the final product would, in this case, only be put through after receipt of an order. Thus the receipt of such an order from a (prospective) customer triggers a sequence of data checks and activities geared toward the timely delivery of the order. Figure 5.5 briefly illustrates, in a flow diagram, the decision making process following the receipt of an order from a customer.

Since the satisfaction of customer order schedules is vital in keeping customer satisfaction, it is very important to make realistic delivery dates. It is far better to promise a safe delivery date than to make an early delivery promise that will not be met. When conditions dictate that an order delivery won't be met, the customer must be

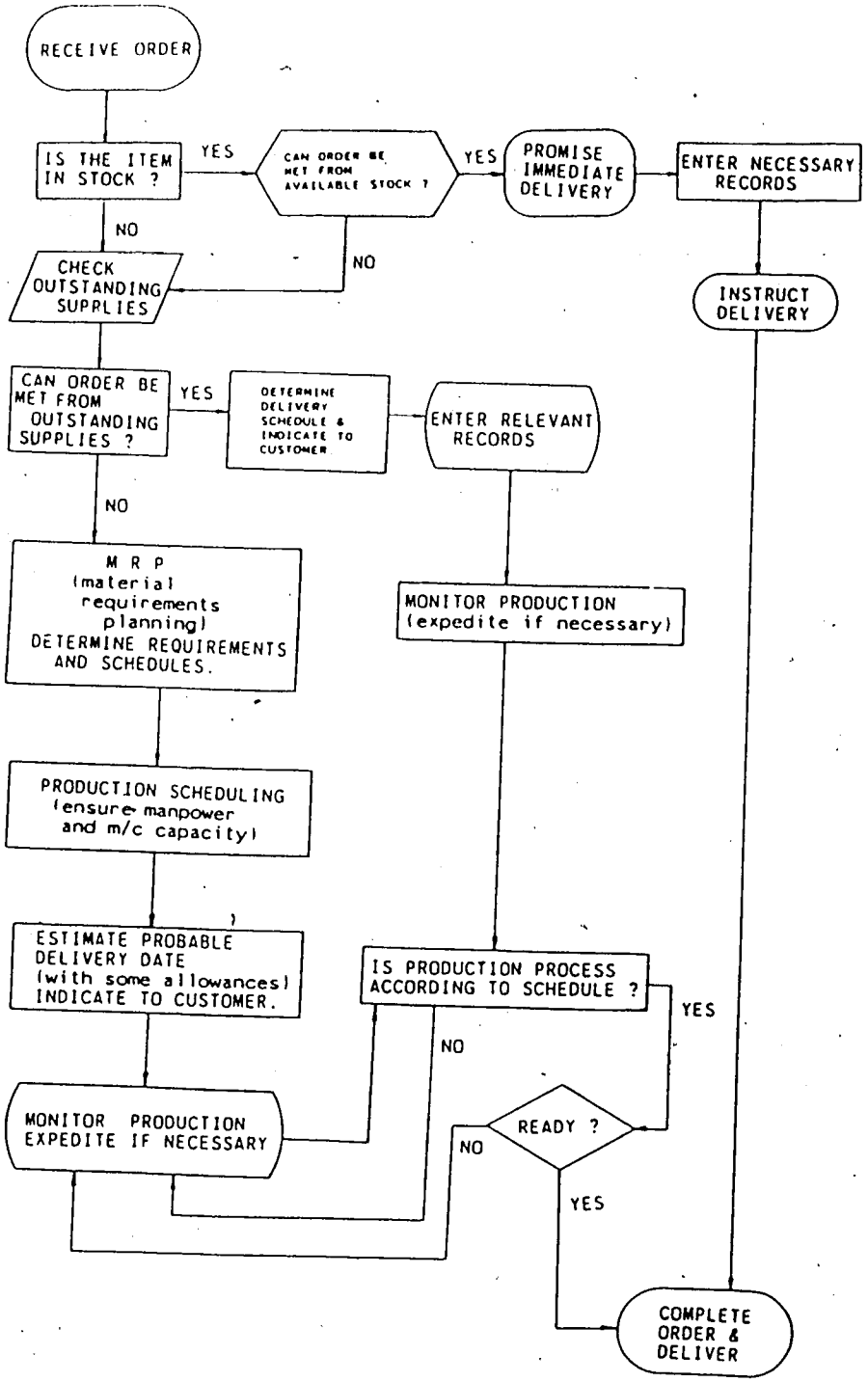


Figure 5.5 The Scheduling Decision Making Process on Receipt of a Customer Order

contacted well in advance and informed of the situation. This will both save the company embarassment as well as show concern to the customer.

Figure 5.5 does not show all the detail involved in the decision making process. Much of the information processing task and the day-to-day monitoring has been compressed to a few blocks in the flow diagram. The purpose of the figure is only to provide a general picture of how the receipt of the order activates the many processes leading for instance to the material requirements planning system, the scheduling system and by implication to the other sub-systems that are linked to these applications.

5.6 Inventory Storage Procedures

The last section of this chapter looks at aspects of inventory storage in the company's warehouse and other inventory location stations within its premises. One notes that while raw materials and finished products are normally stored in the warehouse (or main store), most of the work-in-process and some unused parts are frequently stored in temporary storage elsewhere close to the production departments. A production department may collect a weeks demand of parts etc., which in turn have to be stored in some of these other locations. On the other hand, work-in-process moves around the shop until it comes to its planned final production stage. Only where prolonged delay is anticipated would such work be transferred to a main

storage area.

The inventory control system keeps information with regard to the storage areas of each individual item in the inventory data base. This includes a location code (two characters) and a location name (eight characters). The different inventory locations must be clearly coded/named to provide this vital system information. Like all other system parameters, it is important that this information be kept accurate at all times. Before the location of any inventory item is changed, the applicable new code and name must be substituted in the computer system files accordingly.

Inventory storage should preferably be accomplished in the order of the inventory identification codes. The storage areas should be designed in a manner that is easy for one to trace the location of an inventory item. It is understood of course, that due to size, shape and specific storage requirement variations, storage areas will have to be assorted accordingly. However, for those items that are not stored in their logical location in the main storage area, a tag can be placed around this location indicating the actual location area for this item.

The need to supplement inventory storage information available in the computer system with a systematic arrangement of inventory items in the storage areas arises because it is infeasible to rely on the system whenever one has to pick an item. With the systematic arrangement, store personnel would be able to locate inventory items with ease

without location information from the computer system which may be more usefully engaged with other applications.

It should be noted that neither the location codes nor the location names for inventory locations are unique to the individual inventory items. The location codes and location names only provide information about where the particular item should be stored (or where it can be found), amongst the various locations. Other items (in fact a good number of other items) will have a similar code and location name. This is why the need for a clear storage sequence, with clear labels, is necessary within the storage areas.

The storage sequence will be achieved by allocating the inventory storage bins and/or shelves according to the sequence of inventory identification numbers, so that one can follow the identification number sequence to reach the storage location desired. The exception to this sequence, as earlier mentioned, will be those items which due to their volume, shape or special requirements have to be stored in different areas. Information to this effect would in these cases be placed at their 'logical' location in the main storage area, so that one can be redirected to the appropriate locations.

6. THE INTEGRATED CONTROL SYSTEM: THE SUB-SYSTEM PROGRAMS AND THE IBM COMPUTER SYSTEM

6.1 Introduction

As explained in the abstract, this report is one of three dealing with different sections of the manufacturing control system. The different areas covered were illustrated in Figure 1.2 of this report. This chapter deals first with aspects in linking (integrating) the three sub-system programs into an integrated package, and secondly with the capability of available micro-computer systems (like the IBM package) to handle such a computer package. Specific recommendations with regard to the two will be made at the end of this section.

6.2 Approaches in Intergrating the Programs

The three sub-system programs are in the following specific areas.

1. Demand forecasting, capacity planning and master scheduling - by B.Lim
2. Materials requirements planning - by D.Quek
3. Inventory control, identification and scheduling - this report.

There are two distinct ways in which the three subsystem programs can be assembled.

First, the programs can be integrated so as to directly access one another in the performance of the different

control functions. This implies one main program 'calling' the other(s) as subroutines and thus directly sharing the results of the executions as is the case with any main program and it's subroutines. In order to achieve this the following will be necessary;

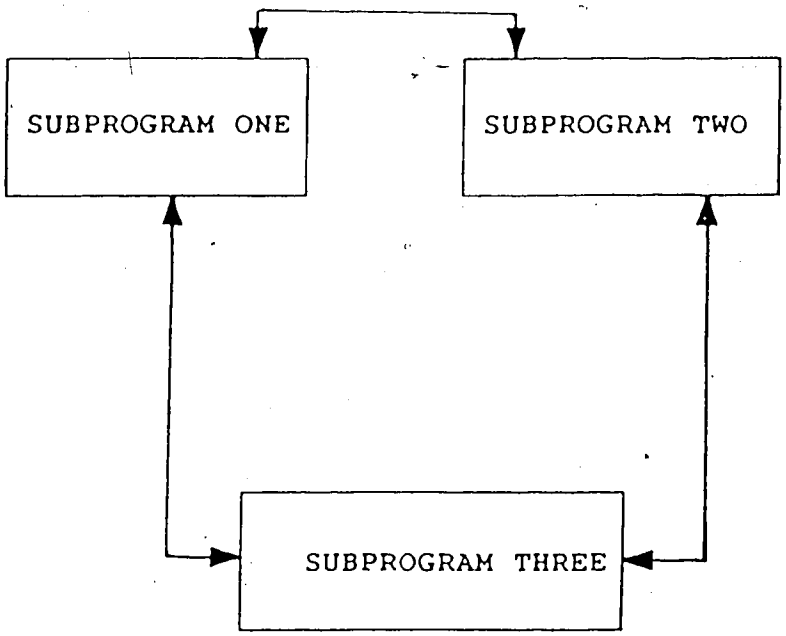
1. Only one main program can be retained in the integrated system since it is not possible to access a main program from another main program. After deciding which program should be the main program, the other main programs should be given appropriate subroutine names and a corresponding calling sequence designed into the resulting program package.
2. The data sharing techniques used in the three program sets require some alteration in order to achieve the needed compatibility. Quek has employed the argument type of data sharing between program units (ie. with transfer types "CALL SUBROUTINE (A, B, C, D) where A,B,C and D are the arguments to be transferred), in the material requirements planning programs. Lim has made use of common blocks (discussed in chapter four) to achieve data sharing between program units in the forecasting, master scheduling and capacity planning programs. Common blocks are also used in the inventory control system programs.

   Common blocks can be inserted in the material requirements planning system programs to provide the required uniformity.

3. A number of different parameter names are used in the subsystem programs to represent the same parameter. If the variable "SOHAND" is used for instance, to stand for "Stock On Hand" in one subsystem program, it should be used likewise in the other programs. If one of the programs were to employ another variable name like "TOTSTK" instead, the transfer of data between the program units would not be accomplished correctly. In the few cases where the parameter names are different, changes should be made accordingly. The variable names used in the inventory control programs were picked so as to conform as far as possible to ones used in the other programs. This will simplify the task involved in making these adjustments.
4. The computer system to be used for the integrated system as well as the program storage devices must have the capacity to provide the necessary memory space for the resulting execution files. This will be elaborated in the discussion of the computer system in the next section. Storage capacity to the level of five hundred to six hundred Kbytes will be required. Although such storage is not provided by the IBM package that is currently in use, it can be achieved through an expansion of the systems core memory and the acquisition of different external storage devices, as will be discussed later. This additional capacity is available on the market.

The first approach in integrating the computer programs is illustrated in Figure 6.1.

The inventory control program can be effectively used as the main program in an integrated program package. The main advantage with this program structure is that it allows for the inventory data to be read into the system before the other subsystems are accessed. One notes that most of the other subsystem applications, if not all, require the use of inventory status information that is available from the inventory control system. If one of the other subsystems were to be used as the main program, it would be necessary to access the inventory control subsystem quite frequently, merely for purposes of reading-in the necessary information. Program efficiency will thus be served better by using the inventory control system as the main program for an integrated production and inventory control system.



NB: The main programs in two of the subprograms are renamed as subroutines to facilitate program control. Thus, subprogram one for instance could retain its mainprogram (this would be the entry point during execution of the integrated package) while the main programs in subprograms two and three are changed into major subroutines.

Figure 6.1 First Alternative Approach for Program Assembly

The second approach is considerably different from the first, and probably more appropriate in a situation where initial investment on the computer system has to be restricted to a system just like the one earlier described. It involves a redesign of the input and output formats of the different programs and a re-definition of the input files/sources for the different executions. This will make it possible for the subsystem programs to use as input the results of the other programs without direct communication. Thus the programs shall be run separately, as they are run at present, except that output that is at present directed only to the user (through the printer or screen) may also need to be placed in diskette files for input to other programs. This approach can be represented as shown in the next figure.

The subsystems in the following figure refer to the following program sections of the integrated production and inventory control system.

Subprogram one: demand forecasting, master production scheduling and capacity planning programs by B. Lim

Subprogram two: material requirements planning system programs by D. Quek

Subprogram three: Inventory control system programs.

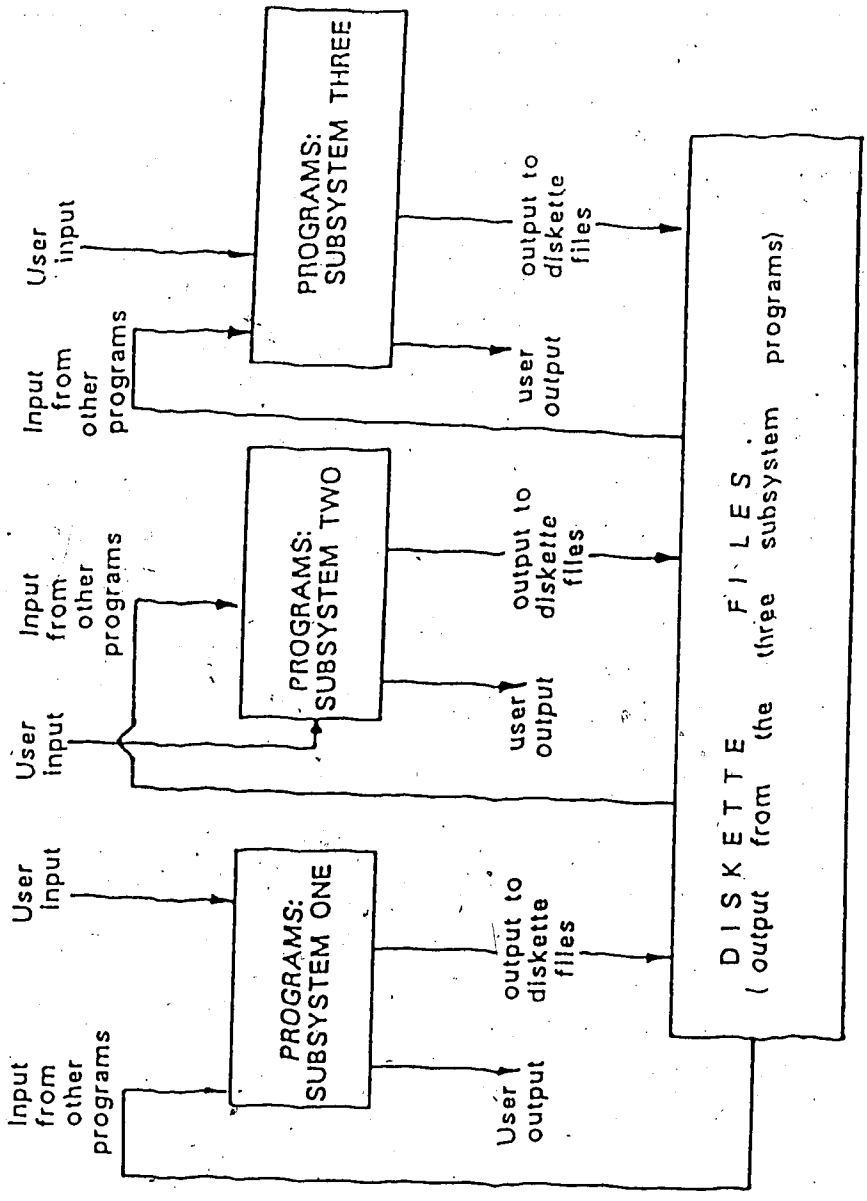


Figure 6.2 Second Alternative Approach for Program Assembly

6.3 The IBM - Micro Computer System

The micro-computer system that has been used for running these programs presently has the following hardware and software components.

1. IBM - system unit : Dual disk drive
   a. 64 K RAM memory (with unit)
   b. 192 K RAM memory (additional, built-in)
   c. 256 K RAM memory (TOTAL)
2. IBM monochrome display
3. IBM / EPSON 80 cps printer
4. Fortran compilers (FOR1, FOR2 & LINK)
5. DOS extended basic
6. Asynchronous communication modem and adapter card
7. Several additional software packages
8. 5 1/4" memory diskettes (160 Kbytes each)

Although the present system components do not provide the kind of capacity that is required to be able to handle the integrated production and inventory control system program (as per first approach described in the previous section), the micro-computer can be expanded to accomplish this capacity requirement.

The area that requires system expansion is with regard to the internal as well as the external (diskette) storage capacity. The inventory control system programs have a combined memory requirement of two hundred Kbytes. If these

programs are integrated with the other two program sets (subsystem one and subsystem two programs as described in section 6.2), the total memory requirement will be between five hundred and six hundred Kbytes.

The system memory, which is at present 256 Kbytes, can be expanded upto one Megabyte of memory by the addition of more memory disks in the disk drive unit. This will more than provide for the memory requirement of the integrated production and inventory control system programs.

As for external storage, large storage devices are now available for micro-computers in the form of hard disks, also called "winchester" disks. According to a University of Alberta Computing Services Bulletin report (April 1983), these disks provide storage capacity in the range of five to twenty Megabytes. The costs for such units are presently put at $2000 to $6000 depending on size. The disk uses a rotating magnetic medium like a floppy diskette but is rigid, providing for a much denser packing of information. The disk also spins faster, allowing for information to be accessed in hundredths or tenths of seconds as opposed to a time of seconds in the case of the floppy diskettes. For purposes of the integrated program, a five Megabyte disk would provide the external storage that is required.

Each of the two approaches described for the eventual integration of the programs has some advantages and disadvantages. Depending on the extent of initial financial commitment the company would like to make, either one of the

two approaches could be considered. The first approach requires a system with much more internal memory and external storage capacity. The second approach can be implemented on a system the size of the one described at the beginning of this section.

With the second approach however, running the programs becomes much more cumbersome since one subsystem program has to be executed and information recorded on diskettes before the next subsystem program can be run. The process of data transfer from the diskettes to each one of the subsystem programs can be quite tedious when large quantities of data are involved.

It is thus apparent that the first integration approach will result in a more effective integration of the computer programs, in as long as the additional investment on computer hardware and software expansion is not a bottleneck.

The next section deals with procedures that are necessary to ensure system security.

6.4 System Security - Implementation of Multiple Security Procedures

In order to ensure the system's integrity, it is important to establish strict procedures to be followed in handling system components and the computer programs.

The computer system, and in particular the data and program storage diskettes, should only be available to a limited number of individuals. Access to the computer programs must be limited to key individuals only. Only a few high-ranking personnel should be permitted to make changes to the system's information and data in the inventory files.

6.4.1 General Security Procedures

The first consideration towards system security is the safe keeping of system components. These include the computer hardware components, software diskettes, program diskettes and any other material used for the inventory control system. The computer system should be placed in a limited access area, if possible in a separate office. Floppy diskettes are rather delicate items requiring careful handling. Diskettes can be easily demaged if carelessly used. Those individuals using the system should be made aware of, and made to follow required safe practices.

Data and program diskettes should be clearly labelled. Back-up copies may be essential for some of the diskettes as a precaution against accidental erasure or damage. Whenever diskettes and other items are not in use, they should be

locked in a secure storage place, out of access from 'non-system' staff.

6.4.2 Program Access Security - Limited Individual Access

Apart from the safety and security measures discussed above, access to the computer programs can be designed so as to allow only particular individuals to access and run different sections of the program. This can be very useful when specific system tasks need to be delegated to only a few system operators. Some program sections may be considered too sensitive to be made available to all system operators. In these cases, the program should have a built-in security system to allow access to specific sections of the program to a few individuals only.

At the moment, the program design allows those with a valid user ID to access any routine in the system. The code (for user identification - ID) entered by the user is verified by the system (by matching the code with codes in a system file), before access to the programs is made. In order to limit access to specific routines, user codes should have built-in significance, identifiable by the system, which will be used by the system to prohibit access to these routines by users whose ID's do not carry the specific code for access to these routines. Whenever these routine are to be executed, the system will first determine whether or not the user is allowed access to the particular routine. If access is not allowed, an error message should

be printed reminding the user that access is not permitted, after which program control should return to the main program.

The above security measures, if strictly followed, will ensure the system's data integrity, and facilitate a long period of trouble free operation of the system.

7. SUMMARY AND CONCLUSIONS

This project has concentrated on inventory identification, inventory control, and shop scheduling and control. The inventory control system is a core subsystem of any integrated production and inventory control system. The other areas of the production and inventory control system, like materials requirements planning and capacity planning, are dependent on the availability of updated information on current inventory status.

The performance of an inventory control system relies on the existence of an effective inventory identification system. The present inventory identification system used by Argo Engineering Ltd. does not provide the coverage and consistency required for a computer-aided inventory control system, as was discussed in chapter three. It is for this reason that the project has dealt with the inventory identification system first, before proceeding to develop the inventory control system.

Aspects of production scheduling, as presented in chapter five, constitute an important functional area in the overall performance of the production and inventory control system. The shop floor is the central focus of company activities, the place where the company's plans and schedules are implemented. Chapter five has concentrated on good record keeping and the proper timing of shop floor operations.

The computer programs for the inventory control system were developed and tested on an IBM micro-computer.

Micro-computers are currently used by a good number of companies and are expected to become even more popular in the coming years. Small and medium size companies like Argo Engineering Ltd. are the ones expected to be most affected by this growing trend. The development of the inventory control program on the IBM micro is in accordance with the anticipated growth in the use of these machines by manufacturing companies, as well as a means to accomplish another goal, that of evaluating the capability of available micro-computer hardware and software. The programs are specifically intended for use on small computers, for which they were designed.

The integration of the different subsystem programs of the integrated production and inventory control system, and the computer capacity requirements for the resulting program package were discussed in chapter six. Recommendations on approaches in integrating the programs and the corresponding considerations are discussed in that section. The information presented in chapter six should provide a good guide during the integration of the subsystem programs. Chapter six has drawn the two main conclusions below:

1. The computer programs for the material requirements planning system, the demand forecasting, capacity planning and master production scheduling system, and the inventory control system can be integrated into one

program package. Only a few modifications are necessary in order to integrate the subsystem programs. These modifications are discussed in section 6.2.

2. Micro-computer packages now available on the market can provide the capacity (internal and external storage) requirements for a production and inventory control system, the size of the one discussed in the previous chapter or even slightly larger.

For a company like Argo Engineering Limited, which has not used a computer in its manufacturing control functions, one is looking at a change from a manual inventory control system (or production and inventory control system) to a computer-aided system. As is the case whenever a new system is introduced, much effort is required in the planning, implementation and monitoring of the new system.

The conversion from a manual inventory control system to a computer-aided system needs to be gradual if the full benefits of a user oriented control system are to be achieved. Among the advantages of such an approach are the following:

1. In the process of conversion, enough time is allowed to incorporate important factors as seen by the users in their day-to-day activities. At the same time, this allows correcting weaknesses in the system before they are too costly and/or overly frustrating.
2. The need for costly support staff in the form of

consultants can be considerably reduced. In a number of cases, excessive use of consultants in the implementation of computer systems makes the systems very expensive and does not provide a self supporting base, perpetuating the need for consultants.

3. It provides the right atmosphere for a company to "grow-up" with its computer system, ensuring that the different computerization stages conform with the existing system requirements - for the best interest of the company both in the short and in the long term.

The assembly of the programs, as discussed in the previous chapter, need only be performed after a thorough performance evaluation of the separate subsystems. Any necessary adjustments must be made at this stage. After one has been satisfied with the different subsystems' evaluation, either of the above assembly approaches, which best suits the requirements or conditions of the company, can then be followed.

The successful implementation of a computer-aided production and inventory control system can only be achieved if the system is carefully designed to suit company requirements and when people are committed to making it work. One would expect that some problems will surface as soon as the system is put into operation. Those using the system should expect some problems and be prepared to correct them when they arise.

In the use of any computer-aided or computerized system, the rule is, never to bypass the computer system, since so doing will render the system's information base inaccurate, thus causing inappropriate actions. Strict procedures must be followed to ensure that any necessary data input to the system is made - at the right time, and that output from the system is acted upon at the earliest convenience. This is the only way effective use of the system can be attained and overall gains in company performance and profits made.

There is reason to believe that the next few years will see even better micro-computing facilities. If this is the case, prospective users of micro-computer systems who will be buying new equipment will have an easier time getting started. Presently available micro-computer hardware and software packages however, can provide the necessary capacity for a production and inventory control system like the one presented in chapter six.

BIBLIOGRAPHY

1. Buffa, E. S. and Miller, J. G. ; Production - Inventory Systems: Planning and Control. Third Edition, Richard Irwin Inc., 1979.

2. Compton, H. K. ; Supplies and Materials Management - A textbook of Purchasing and Supply: Macdonald and Evans Press, 1979.

3. Dennis, Q. ; Computerized Material requirements Planning. Unpublished M.Eng. project report - University of Alberta, Edmonton - 1982.

4. Fuchs, J.H.; Computerized Inventory Control systems; Prentice-Hall Inc.,1978

5. Greene, J. H.(editor); Production and Inventory Control Handbook, McGrawHill Book Company - 1970.

6. Greene, James H.; Production and Inventory Control: Systems and Decisions. Richard Irwin Inc. 1974

7. Hopeman, Richard J.; *Production*- Concepts, Analysis and Control. Charles E. Merrill Co. 1976

8. IBM- Personal Computer ; Computer Language Series Manuals; Disk Operating System & Fortran Compiler.

9. Johnson, George A.; "Reviewing Part Coding and Classification" *Production and Inventory Management Review* (December 1982):34-38

10. Lim, B. F. ; A Manufacturing Resource Planning System. Unpublished MSc. thesis, University of Alberta - 1982.

11. Mabert, V. A. and Moodie, C. L. ; Production Planning, Scheduling and Inventory Control: Concepts, Techniques and Systems. Industrial Engineering Press - IIE -1982.

12. MacConnell, W. ; Classification and Coding - An Introduction and Review of Classification and Coding Systems. *British Institute of Management* - 1971

13. Meldman, McLeod, Pellicore and Squire ; *RISS* - A Relational Data Base Management System for Minicomputers. Computer Science Series - 1978

14. Moore, F.G. and Hendrick, T.E.; Production/ Operations Management - Eighth Edition, Richard Irwin Inc. 1980

15. Schonberger, Richard J.; Casting Off Old P&IC Models. *Production and Inventory Management Review* - March 1983.

16. Smith, G. W. ; Engineering Economy: Analysis of Capital Expenditures. Iowa State University Press - 1979.

17. Steffy, W. et al; Inventory Control for the Small and Medium Sized Firm. Computer Aided Inventory Control and Manual Inventory Control Systems - The University of Michigan, 1970

18. Tsichritzis, D.C. and Lochovsky, F.H.; Data Base Management System - Computer Science and applied Mathematics Series, Academic Press - 1977

19. Wallace, Thomas F.; *Dictionary:* American Production and Inventory Control Society - 1980

20. Wild, Ray ; Production and Operations Management: Techniques and Principles. Holt, Rinehart & Winston - 1979.

APPENDIX A - Main Variables used in the Computer Programs

AVDEM	...Average annual demand for the item.
CLASS	...Inventory category, either A,B or C classification.
COD(I)	...Inventory identification, Ith character
CODLOC	...Location code for the inventory item
COST1	...Inventory unit cost
COSTOD	...Cost for placing an order for materials
DATE	...The current date - day of program run
DATE1	...The first date a particular inventory item is used.
DATE2	...The last date a transaction was made on the item
DATEA	...Integer representation of current date(mo/dy) -I4
DATEB	...Integer representation of current date(mo/dy/yr) format=I6
DEPCOD	...User department code for inventory item
EOQTY	...Economic order quantity
EXPTAV	...Expected arrival date for a given order
HOLCOS	...Inventory holding cost $/item/year
ICALNO	...Subroutine identification code, for major subroutines only
INIT	...User's initials or name - 8 characters
INVLEV	...Inventory level code, 0,1,2,3 etc.
IRECNO	...The inventory record number in the database
ITEREC	...An array of inventory record numbers, used to produce reports for specific inventory items from this array.
MOB	...Make or Buy option for inventory
MODE	...Mode of entry into a subroutine - direct from the main program or from another subroutine either 1 or 2.
NAME	...Inventory description (20 characters)
NTOTAL	...Total number of array elements in the inventory record number array ITEREC
ORDRNO	...The serial number of an order (order 1,2 or 3)
ORDRDT	...The date of placement of a given order
ORDRQT	...Quantity of an order
PAR	...A title parameter for producing different titles (upto 36 characters long).
PARTNO	...Inventory identification code
PLDTIM	...Procurement lead time, either for buying or production.
ROLEV	...Reorder level for inventory item
SEVLEV	...Customer service level - probability of meeting orders in time (ie. no stock-outs)
SOHAND	...Stock quantity on hand
UNIT	...The unit of measure for the inventory item, eg. EA,lb,kg,ton etc.
USERCD	...Program user identification code

APPENDIX B - Program Units Used in the Inventory Control System

CHGREP ...Transaction/Change report subroutine
DALTER ...Subroutine makes necessary alterations following different transactions
DATPRO ...Data processing/ parameter evaluation subroutine
DATVER ...Inventory identification verification subroutine
DREAD ...Subroutine for reading inventory data from the database
DWRITE ...Subroutine for writing inventory data update into the database
ENQURY ...Inventory information enquiry subroutine - gives the user all information pertaining to that item
ENTRES ...User/date input routine. Also checks user ID
EXCEPT ...Exception report generating subroutine
INTROD ...Program execution: introductions to the user. Tells the user which section of the program is in progress as execution continues.
MAIN11 ...Main program number one - major transactions etc.
MAIN22 ...Main program number two - parameter evaluation, regular report generation etc.
MAIN33 ...Main program number three. Inventory identification and sorting procedures
NEWDAT ...New inventory items' information register routine
OPTION ...Program access options subroutine. Provides the user with the necessary information to be able to access the appropriate program/subroutine.
ORDERS ...This is the order processing subroutine
ORDSLP ...Order placement and/or order receipt documentation and processing subroutine
PROMPT ...Subroutine prompts the user to indicate the kind of transactions he/she wants to make
REGREP ...Regular report generation subroutine - accessed only at the option of the user
RTHEAD ...Report title printout subroutine. Depending on the value of PAR(the title) when accessed, the appropriate title is printed
SORTIT ...Inventory- type sorting by identification code characters. The sorting can be performed for any combination of significant code characters
THEAD ...Company title head subroutine - includes date
TRANSN ...This is the general transactions subroutine
VARCOD ...This subroutine lists the inventory variables that can be altered through the various transactions

APPENDIX C - Listing of Source Programs

```
$TITLE:'PRODUCTION & INVENTORY CONTROL SUBSYSTEM'
$SUBTITLE:'GENERAL TRANSACTION SUBROUTINES'
$PAGESIZE:60

      PROGRAM MAIN11
C  ************************************************************************
C  **                                **
C  **  THIS MAIN PROGRAM HANDLES FOUR [4] MAJOR SUBROUTINES. THE 1st **
C  **  IS THE MAIN TRANSACTIONS SUBROUTINE, THE SECOND IS THE ENQU-  **
C  **  -IRY SUBROUTINE, THE THIRD IS THE ORDER PROCESSING ROUTINE,   **
C  **  AND THE LAST ONE IS THE SORTING SUBROUTINE. TWO OTHER PROGRAMS**
C  **                    HANDLE OTHER SUBROUTINES..                  **
C  **
C  ************************************************************************
C
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &              AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB, CLASS,DEPCOD,CODLOC,LOCNAM
        COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO, YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &              EXPTAV,ORDRNO,ORDRDT,ORDRQT

C....NB. THE DIFFERENT VARIABLES USED IN THE PROGRAM
C     ARE DESCRIBED IN AN APPENDIX OF THE REPORT..
C
5     CONTINUE
      DO 8 I=1 10
        MM(I)=0
8     CONTINUE
```

```
      NO='N'
      YES='Y'
      ICALNO=1
C ....THE SUBROUTINES INTROD, ENTRES AND EXCEPT ARE
C     ACCESSED AT THIS POINT.
C

      CALL INTROD
C     ***********
      CALL ENTRES
C     ***********

         IF(MM(4).EQ.3) GOTO 500
      CALL EXCEPT

C........THE OPTION SUBROUTINE IS USED HERE TO PROVIDE
C      THE AVAILABLE EXECUTION OPTIONS TO THE USER..
C
20    CONTINUE
      CALL OPTION
      MODE=1
      MM(5)=10

C ........EITHER ONE OF THE FOLLOWING OPTIONS IS
C        EXECUTED NEXT, DEPENDING ON THE USER'S
C        OPTION CHOICE.
C
      IF( MM(1).EQ.1) THEN
            CALL TRANSN
          ELSEIF( MM(1).EQ.2) THEN
            CALL ENQURY
          ELSEIF( MM(1).EQ.3) THEN
            CALL ORDERS
C         ELSEIF( MM(1).EQ.4) THEN   >>MOVED TO 2nd
C            CALL DATPRN             >>PROGRAM !!!
          ELSEIF( MM(1).EQ.5) THEN
            CALL SORTIT
      ENDIF

      IF( MM(1).EQ.9) GOTO 500

400   WRITE(*,420)
C .........BACK FROM THE SUBROUTINE ONE MAY RE-ENTER THE
C     MAIN PROGRAM IF ANOTHER ROUTINE IS REQUIRED......
C
420   FORMAT('0','DO YOU LIKE TO RE-ENTER THE MAIN PROGRAM?? Y/N')
      READ(*,'(A)',ERR=400) ANS
      IF(ANS.EQ.YES) GOTO 20

500   CONTINUE
      WRITE(*,540) INIT
540   FORMAT('0','THANK YOU ','[[ ',A,' ]]',' FOR USING THIS ',
     &'PROGRAM.',/,'0',50X,'E.M.KOMBE [83]',/,' ',48X,18('='),/,' ',/)
```

```
      CONTINUE
      END


      SUBROUTINE ENTRES

C  ***********************************************  ******************************
C  **                                                                             **
C  **  THIS SUBROUTINE INPUTS THE DATE, USER NAME, USER CODE AND  **
C  **  CHECKS THAT THE USERCODE ENTERED IS ACCEPTABLE.     **
C  **                                                                             **
C  *********************************************************************************

      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
            COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODENO,USERCD
      CHARACTER*10 AB
      CHARACTER*8 DATE,INIT,PARTNO
      CHARACTER*36 PAR
      INTEGER DATE1,DATE2,DATEA,DATEB

      WRITE(*,5)
5     FORMAT('0','THE DEPARTMENT OF MECHANICAL ENGINEERING-- ',
     &'UNIVERSITY OF ALTA.',/,63('='),/,'0',' ')
C .....THE USER'S NAME, DATE AND USER ID IS ENTERED
C     THE SYSTEM CHECKS FOR THE EXISTENCE OF THE GIVEN
C     ID IN THE USERCODE FILE.
C
10          WRITE(*,20)
20    FORMAT('0','PLEASE ENTER YOUR INITIALS','.' [8 CHARACTERS]')
      READ(*,30,ERR=10) INIT
30    FORMAT(A)
40    WRITE(*,50)
50    FORMAT('0','ENTER TODAY''S DATE [M/D/Y -eg.04/22/83]')
      READ(*,30,ERR=40) DATE
      MM(4)=1

60    WRITE(*,70)
70    FORMAT('0','ENTER YOUR PERSONAL USER ID. CODE')

      READ(*,80,ERR=60) USERCD
80    FORMAT(A)

      OPEN(8,FILE='USERCOD.INV')
90    READ(8,'(A)',END=100) CODENO
      IF(CODENO.EQ.USERCD) GOTO 110
      GOTO 90

100   IF( MM(4).EQ.3) GOTO 110
```

```
      MM(4)=MM(4)+1
      PAUSE 'INVALID USER CODE - TRY AGAIN!'
      GOTO 60

110   CONTINUE
C.....THE DATE IS BROKEN DOWN TO SIMPLE, DIGIT
C.....FORM VARIABLES FOR RECORDKEEPING PURPOSES.
C
      WRITE(AB,'(A)') DATE
      READ(AB,120) I,J,K
120   FORMAT(I2,1X,I2,1X,I2)
      WRITE(AB,'(3I2)') I,J,K
      READ(AB,'(I6)') DATEA
      READ(AB,'(I4)') DATEB
      WRITE(*,130) DATEA,DATEB
130   FORMAT('0','RECORDED DATES IN DATABASE==',I6,' & ',I4,
     &' [m/d/yr]')

      CONTINUE
      END


      SUBROUTINE DATVER

C  ************************************************************************
C  **                                                                    **
C  **  THIS SUBROUTINE VERIFIES THE EXISTENCE OF ITEM INFORMA-   **
C  **  -TION IN THE DATA BASE. WHEN ACCESSED FROM THE TRANSAC-   **
C  **  -TIONS SUBROUTINE IT ALSO CALLS *NEWDAT* FOR INPUTTING    **
C  **  INFORMATION FOR A NEW ITEM:                                       **
C  **                                                                    **
C  ************************************************************************
      COMMON /C1/ ICALNO, MM(10), MODE, IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,PARTMM,DATE,INIT
      CHARACTER*36 PAR

      OPEN(7,FILE='PARTNOS.INV',ACCESS='DIRECT',RECL=8)
      ANS='P'
      I=1

C.....THE PART NUMBER ENTERED IS VERIFIED FROM INVENTORY
C.....RECORDS. IF NECESSARY THE NEW-DATA SUBROUTINE IS
C     CALLED IN TO REGISTER NEW INVENTORY INFORMATION.
C
10    READ(7,20,REC=I,END=30) PARTMM
      IF(PARTMM.EQ.PARTNO) THEN
          IRECNO=I
          CALL DREAD
      ELSE
          I=I+1
```

```
            GOTO 10
         ENDIF

         GO TO 60
20       FORMAT(A)
30       WRITE(*,50)
50       FORMAT('0','THIS IS A NEW ITEM - OK?  Y/N')
               READ(*,'(A)') ANS
               IF(ANS.NE.NO) THEN
                  IRECNO=I
                  MM(6)=66
                  IF( MM(5).EQ.4) CALL NEWDAT
               ENDIF
60       CONTINUE
         END



         SUBROUTINE OPTION

C  ************************************************************
C  **  THIS SUBROUTINE PRINTS THE VARIOUS OPTIONS  **
C  **  AVAILABLE IN THE TRANSACTIONS MAJOR PROGRAM **
C  ************************************************************

         COMMON /C1/ ICALNO,MM(10),MODE,IRECNO


         ICALNO=3
            CALL INTROD
         WRITE(*,10)
10       FORMAT('0','THE FOLLOWING SUBROUTINES ARE CURRENTLY AVAILABLE',
        &/,' ','PLEASE ENTER ONE OF THE RELEVANT OPTIONS DEPENDING ON ',
        &'THE INTENDED',/,' ','USE OF THE SYSTEM.',
        &/,'0','OR ENTER NINE [9] TO EXIT!')

C.........THE DIFFERENT AVAILABLE OPTIONS ARE LISTED
C.........FROM THIS ROUTINE.
C
20       WRITE(*,30)
30       FORMAT('0',' 1....GENERAL TRANSACTIONS, INPUT/OUTPUT ETC.',/,
        &' ',' 2....ENQUIRY===> ANY DATA OR PARTICULAR INFORMATION.',/,
        &' ',' 3....ORDERS,--PROCESSING OF ORDERS, PLACEMENT/RECEIPT',/,
        &' ',' 4....DATA PROCESSING. [*MOVED TO PROG2 !]  PARAMETER ',
        &'EVALUATION..',/,
        &' ',' 5....SORTING FOR PART ID OF GIVEN SIGNIF. CODES:',/,
        &'0','ENTER THE APPROPRIATE OPTION:',/,'0','ENTER 9 TO EXIT !!')

         READ(*,40,ERR=20) II
         MM(1)=II

40       FORMAT(I1)
         END
```

```
      SUBROUTINE TRANSN

C  ***************************************************************************
C  **                          **
C  **  THIS IS THE MAIN TRANSACTIONS SUBROUTINE. ALL STOCK **
C  **  ENTRIES AND GENERAL DATA CHANGES ARE EXECUTED FROM THIS **
C  **  SUBROUTINE. IN THIS PROCESS IT EMPLOYS MANY OF THE OTHER **
C  **  ROUTINES. A RECORD IS KEPT OF ALL TRANSACTIONS MADE TO   **
C  **  FACILITATE THE PRODUCTION OF A TRANSACTION REPORT AT THE **
C  **                END OF THE EXECUTIONS:     **
C  **                          **
C  ***************************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
      COMMON /C7/ ITEREC(100),NTOTAL

      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT,BLANK
      CHARACTER*36 PAR
      INTEGER DATE1,DATE2,DATEA,DATEB

      BLANK='        '
      ICALNO=4
      MODE=1
          CALL INTROD

          MM(8)=0

2     WRITE(*,3)
3     FORMAT('0','IN THE FOLLOWING PROMPT ENTER BLANKS FOR THE PART',
     &/,' ','NUMBER IF YOU WANT TO EXIT THE TRANSACTIONS SUBROUTINE.')
5     PAUSE 'NEW PART NUMBERS MAY BE INPUT !'
        MM(2)=24
        MM(3)=1
C.....
C.....THE USER IS PROMPTED FOR AN ENTRY BY THE
C      THE SUBROUTINE "PROMPT" WHICH FOLLOWS.
C
10    CALL PROMPT

      READ(*,20,ERR=10) PARTNO
20    FORMAT(A)

      IF(PARTNO.EQ.BLANK) GO TO 100
            MM(5)=4
C.....
C.....NEXT THE DATA VERIFICATION SUBROUTINE IS ACCESSED.
C
      CALL DATVER
```

```
22    CONTINUE
      IF(MM(6).NE.66) GOTO 25
      WRITE(*,24) PARTNO
24    FORMAT('0','ARE YOU MAKING MORE CHANGES TO PART # ',A,' ?')
      READ(*,'(A)',ERR=22) ANS
         IF(ANS.EQ.NO) GOTO 2
25    WRITE(*,30) IRECNO
30    FORMAT('0','INVENTORY RECORD NUMBER=',I4,/,' ',30('='))

      MM(6)=0
      MM(8)=MM(8)+1
      NTOTAL=MM(8)
      ITEREC(NTOTAL)=IRECNO

35    CALL VARCOD
C     ***********
C.....THE SUBROUTINE VARCOD IS EMPLOYED TO ENABLE THE
C     USER MAKE A PICK ON THE PARAMETER(S) TO BE UPDATED.
C

40    WRITE(*,50) PARTNO
50    FORMAT('0','ENTER THE CODE# FOR THE DESIRED TRANSACTION [I2]',
     &/,' ','ON THE ITEM NUMBER ',A)
      READ(*,60,ERR=40) II
      MM(2)=II
60    FORMAT(I2)

      CALL DALTER
C     ************
C.....THIS SUBROUTINE IS CALLED TO EFFECT THE REQUIRED
C     PARAMETER CHANGES..
C
      DATE2=DATEA
      IF( MM(2).NE.0) GOTO 35

      CALL DWRITE
C     ***********
C.....AT THE END OF THE CHANGES THE SUBROUTINE DWRITE
C     IS USED TO WRITE THE UPDATED FIGURES IN DATABASE.
C
      GO TO 2
100   CONTINUE

      WRITE(*,110)
110   FORMAT('0','DO YOU WANT A TRANSACTION REPORT ? Y or N ?',/,
     &' ','PLEASE NOT THAT AFTER EXITING S/ROUTINE &/OR PROGRAM',/,
     &' ','YOU MAY NOT BE ABLE TO PRODUCE THIS REPORT !')
      READ(*,'(A)',ERR=100) ANS
      IF(ANS.EQ.YES) THEN
C.......
C.......BEFORE EXITING THE TRANSACTIONS SUBROUTINE THE
C      USER HAS THE OPTION TO PRINT A TRANSACTINS REPORT.
C
        CALL CHGREP
```

```
      ELSE
         WRITE(*,120)
120      FORMAT('0','REALLY ? NO?  RE-ENTER [N] TO CONFIRM !')
         READ(*,'(A)') ANS
         IF(ANS.EQ.YES) CALL CHGREP
      ENDIF
      CONTINUE
      END


      SUBROUTINE THEAD

C  ***************************************************************
C  **                       **
C  **  THIS SUBROUTINE PRINTS THE COMPANY TITLE HEAD     **
C  **  AND DATE FOR VARIOUS REPORTS THAT ARE GENERATED.  **
C  **                       **
C  ***************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

      IF(MODE.EQ.2) OPEN(2,FILE='PRN')
      IF(MODE.EQ.1) OPEN(2,FILE='USER')

      WRITE(2,10) DATE
10    FORMAT('0',' ',/,'0',36('MW'),/,' ','MWM',66X,'MWM',/,' ',
     &'MWM',3X,'ARGO ENGINEERING LIMITED.',20X,'Date:(mo/da/yr)',
     &3X,'MWM',/,' ','MWM',3X,25('='),20X,15('+'),3X,'MWM',/,' ',
     &'MWM',3X,'(Specialists in Material Handling Equipment.)',6X,
     &A,4X,'MWM',/,' ','MWM',66X,'MWM',/,' ',36('WM'),/,'0',' ')
      END


      SUBROUTINE RTHEAD

C  ***************************************************************
C  **  THIS SUBROUTINE PRINTS THE APPROPRIATE HEADING "PAR"**
C  **  FOR DIFFERENT OUTPUTS AS REQUIRED IN THE PROGRAMS.  **
C  ***************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO, DATE,INIT
      CHARACTER*36 PAR
```

```
      IF(MODE.EQ.2) OPEN(2,FILE='PRN')
      IF(MODE.EQ.1) OPEN(2,FILE='USER')

*     WRITE(2,10)
10    FORMAT('0',' ',/,' ','    ')
C.....
C.....THE VARIABLE PAR IS THE NAME OF THE REQUIRED TITLE..
C
      WRITE(2,20) PAR
20    FORMAT('0',10X,50('#'),/,11X,5('#'),40X,5('#'),/,11X,5('#'),
     &2X,A,2X,5('#'),/,11X,5('#'),40X,5('#'),/,11X,50('#'),/,'0',' ')
     &/,' ',8X,'II',50('_'),'II',/)
      WRITE(2,30)
30    FORMAT('0',' ',/)
      CONTINUE
      END
      SUBROUTINE ORDSLP(I,J)

C  ***********************************************************************
C  **                        **
C  **  THIS SUBROUTINE IS USED TO PRODUCE ORDER PLACEMENT      **
C  **  AND/OR ORDER RECEIPT SLIPS. IT IS ACCESSED FROM THE     **
C  **        MAIN SUBROUTINE *ORDERS*.     **
C  ***********************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*10 VEND4
      CHARACTER*20 NAME,VEND1,VEND2,VEND3
      CHARACTER*36 PAR

      INTEGER EXPTAV,ORDRNO,ORDRDT,ORDRQT

      OPEN(2,FILE='PRN')
         MODE=2
         CALL THEAD

      IF(I.EQ.1) PAR='M A T E R I A L    O R D E R    SLIP'
      IF(I.EQ.2) PAR='M A T E R I A L   R E C E I PT  SLIP'

      CALL RTHEAD
      IF(I.EQ.1) WRITE(2,20)
      IF(I.EQ.2) WRITE(2,30)

20    FORMAT('0','T O :',/,' ','======',/)
30    FORMAT('0','F R O M :',/,' ','==========',/)
```

```
 40     WRITE(*,50) PARTNO
 50     FORMAT('0','ENTER VENDOR/SUPPLIER CODE FOR THE ITEM NO. ',A)
        READ(*,60,ERR=40) IVEN
 60     FORMAT(BN,I3)

        OPEN(9,FILE='VENDSUP.INV',ACCESS='DIRECT',RECL=72)
        READ(9,70,REC=IVEN,END=90) VEND1,VEND2,VEND3,VEND4
        GO TO 130
 70     FORMAT(4A)
 90     WRITE(*,100)
 100    FORMAT('0','INVALID VENDOR CODE. CHECK PROPER CODE.',/,' ',
      &'WOULD YOU LIKE TO RE-INPUT THE CODE?   Y/N')
        READ(*,'(A)',ERR=90) ANS
        IF(ANS.NE.NO) GOTO 40
        PAUSE 'PLEASE SEE NOTE ON CANCELLED SLIP !!'

        WRITE(2,110)
 110    FORMAT('0',10X,'======I N V A L I D=========',/,'0',
      &13X,'C A N C E L L E D',/,'0','Important Note:',/,' ',15('*'),
      &/,5X,'Please note that this order had been enterd in database.',
      &/,' ',5X,'You should re-enter subroutine to cancel the order!')
        GOTO 200
 130    WRITE(2,140) VEND1,VEND2,VEND3,VEND4
 140    FORMAT('0',5X,A,/,6X,A,/,6X,A,/,6X,A,/,6X,10('-'))
        WRITE(2,150)
 150    FORMAT('0','ORDER NUMBER',10X,'OF DATE[mo/dy]',10X,'QUANTITY',
      &/,' ',12('='),10X,14('='),10X,8('='))
        WRITE(2,160) ORDRNO(J),ORDRDT(J),ORDRQT(J),UNIT
 160    FORMAT('0',5X,I5,20X,I4,14X,I5,4X,'##',A,'##')
        WRITE(2,170) NAME
 170    FORMAT('0','.',/,' ','MATERIAL DESCRIPTION:',/,' ',21('='),
      &/,' ',20X,A)
        WRITE(2,180) INIT
 180    FORMAT('0',' ',/,' ',58X,'by:',2X,A,/,62X,8('-'),/,45X,
      &'for Argo Engineering Limited.')
 200    CONTINUE
        END


        SUBROUTINE INTROD

C  ***************************************************************************
C  **                          **
C  **  THIS SUBROUTINE IS DESIGNED TO OFFER AN INTRODUCTORY **
C  **  NOTE TO THE USER OF THE PROGRAMS REGARDING THE TYPE  **
C  **  OF FUNCTION BEING ACCESSED. THIS IS A VERY GOOD AID IN **
C  **  ENSURING THAT THE USER CAN EASILY FOLLOW THE EXECUTION. **
C  **                          **
C  ***************************************************************************
        COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
        COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
```

```
      CHARACTER*1 ANS, NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

      IF(ICALNO.EQ.1) DATE='%%%%%%%%'
10    WRITE(*,20) DATE
20    FORMAT('0',10X,50('*'),/,' ',10X,'**',46X,'**',/,' ',10X,
     &'**',2X,'ARGO ENGINEERING LIMITED',6X,'DATE:(M/D/Y)',2X,'**',
     &/,' ',10X,'**',37X,A,1X,'**')
      GOTO(40,50,60,70,80,90,100,110,120,130,180) ICALNO
      GO TO 180
      CONTINUE
40    PAR='   MAIN   TRANSACTION   PROGRAM  '
      GOTO 200
50    PAR=' EXCEPTION   REPORTING  SUBROUTINE  '
      GOTO 200
60    PAR='  TRANSACTION   OPTIONS   LISTING   '
      GOTO 200
70    PAR='INPUT/OUTPUT AND ALTERATIONS ROUTINE'
      GOTO 200
80    PAR=' DATA & INFORMATION ENQUIRY ROUTINE '
      GOTO 200
90    PAR='ORDER PLACEMENT/PROCUREMENT ROUTINE '
      GOTO 200
100   PAR=' DATA EVALUATION/PROCESSING  ROUTINE'
      GOTO 200
110   PAR='NEW ITEM INFORMATION INPUT SUBROUTIN'
      GOTO 200
120   PAR='REGULAR REPORT GENERATING SUBROUTINE'
      GOTO 200
130   PAR='   VARIABLE   CODE   LISTING        '
      GO TO 200
      CONTINUE
180   PAUSE 'UNDEFINED PROGRAM NAME  ..INTERNAL..'
      GOTO 300
200   WRITE(*,220) PAR
220   FORMAT(' ',10X,'**',2X,'THIS IS THE:',/,' ',10X,'**',2X,
     &'===>>',' ',A,2X,'**',/,' ',10X,'**',46X,'**',/,' ',10X,
     &50('*'))

300   CONTINUE
      END


      SUBROUTINE PROMPT

C  ************************************************************************
C  **   THIS SUBROUTINE IS USED BY VARIOUS OF THE PROGAM SECTIONS *
C  **   TO PROMPT THE USER FOR DIFFERENT KINDS OF INFORMATION     *
C  **   AS REQUIRED BY THE PARTICULAR PROGRAM SECTION.     ********
C  ************************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
```

```
      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

5     J=MM(2)-10
10    GO TO(20,20,40,50,60,70,80,90,100,650,650,110,120,130,
     &140,150,160,170,180,650,650,650,190,200,210,220,230,240,250,260,
     &270,280,290,300,310) J

      GOTO 650

20    PAR=' TOTAL AMOUNT (SOHAND); [F8.2]      '
      GOTO 700
40    PAR='HOLDING COST/ITEM/YR(HOLCOS) [F6.2] '
      GOTO 700
50    PAR=' ORDERING COST -(COSTOD), [F6.2]    '
      GOTO 700
60    PAR='INVENTORY UNIT COST (COST1) , [F8.2]'
      GOTO 700
70    PA   ' SERVICE LEVEL (SEVLEV), [F4.2]    '
      GOTO 700
80    PAR='ECON. ORDER QUANTITY(EOQTY), [I5]   '
      GOTO 700
90    PAR=' RE-ORDER LEVEL (ROLEV), [I5]       '
      GOTO 700
100   PAR=' PROCUREMENT LEAD TIME(PLDTIM) [I4] '
      GOTO 700
110   PAR=' AVERAGE ANNUAL DEMAND(AVDEM), [I8] '
      GOTO 700
120   PAR=' INVENTORY LEVEL(S)  (INVLEV), [I3] '
      GOTO 700
130   PAR=' INVENTORY IDENTIFICn #(PARTNO)-[A8]'
      GOTO 700
140   PAR=' INVENTORY DESCRIPTION-(NAME), [A20]'
      GOTO 700
150   PAR=' INVENTORY UNIT MEASURE-(UNIT),[A4] '
      GOTO 700
160   PAR=' MAKE OR BUY CASE-(MOB),  [A1]      '
      GOTO 700
170   PAR=' INVENTORY CLASS -(CLASS),  [A1]    '
      GOTO 700
180   PAR=' USER DEPTMENT CODE (DEPCOD), [A5]  '
      GOTO 700
190   PAR=' INVENT. ENTRY CODE (DATE1),  [I6]  '
      GOTO 700
200   PAR=' LAST INVENT. USE DATE (DATE2), [I6]'
      GOTO 700
210   PAR=' INVENTORY LOCATION CODE(CODLOC)[A2]'
      GOTO 700
220   PAR=' INVENT. LOCATION NAME (LOCNAM)[A8]'
      GOTO 700
230   PAR=' ORDER NO 1 SERIAL #(ORDRNO(1)--I5) '
```

```
      GOTO 700
240   PAR=' ORDER NO I DATE (ORDRDT(I)-m/d--I4 '
      goto 700
250   PAR=' ORDER QUANTITY, ORD I(ORDRQT(I)-I5)'
      GOTO 700
260   PAR=' EXP. ORDER RECEIPT (EXPT(I))m/d,-I4'
      GOTO 700

270   GOTO 650
280   GOTO 650
290   GOTO 650
300   GOTO 650
310   GOTO 650

650   PAUSE 'INVALID PARAMETER CODE ! ???'
      GOTO 800
700   CONTINUE
      IF( MM(3).EQ.1) WRITE(*,710) PAR
      IF( MM(3).EQ.2) WRITE(*,720) PAR

710   FORMAT('0','PLEASE (RE)*ENTER  THE ',A)
720   FORMAT('0',' THE ', A,' IS')

800   CONTINUE
      END
      SUBROUTINE VARCOD

C  ***********************************************************************
C  **                                                                   **
C  **  THIS SUBROUTINE PROVIDES A LISTING OF THE DIFFERENT  **
C  **  PARAMETERS THAT MAY BE ALTERED OR LISTED FROM THE DATA  **
C  **  THAT IS KEPT BY THE SYSTEM. IT IS CALLED FROM VARIOUS  **
C  **  OTHER SUBROUTINES DURING DIFFERENT EXECUTION PROCESSES.    **
C  **                                                                   **
C  ***********************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO

      ICALNO=9
        CALL INTROD

      WRITE(*,30)
30    FORMAT('0','  ENTER TWO BLANKS (00) TO EXIT !')
      WRITE(*,40)
40    FORMAT('0','11...WITHDRAWAL OF INVENTORY or returns(-ve)',/,
     &' ','12...INVENTORY RECEIPTS - ORDER ARRIVALS ONLY !',/,' ',
     &'13...HOLDING COSTS (HOLCOS)',/,' ','14...ORDERING COSTS (COST',
     &'OD)',/,' ','15...INVENT. ITEM COST (COST1)',/,' ','16...SERVICE'
     &,' LEVEL (SERLEV)',/,' ','17...ECONOMIC ORDER QUANTITY(EOQTY)',/,
     &' ','18...RE-ORDER LEVEL(ROLEV)',/,' ','19...PROCUREMENT LEAD',
     &' TIME (PLDTIM)',/,' ','22...AVERAGE ANNUAL DEMAND (AVDEM)',/,
     &' ','23...INVENTORY LEVEL(S), (INVLEV)',/,'0',' ')

      PAUSE ' ***** PICK THE RELEVANT CODE *****'
```

```
      WRITE(*,60)
60    FORMAT(' ','24...CHANGE OF PART NUMBER (PARTNO)',/,' ',
     &'25...PART DESCRIPTION--NAME',/,' ','26...INVENTORY MEASURE',
     &' UNIT',/,' ','27...MAKE OR BUY  (MOB)',/,' ','28...INVENTORY',
     &' CLASS (A/B/C)',/,' ','29...USER DEPARTMENT CODE (DEPCOD)',
     &/,' ','35...LOCATION CODE (CODLOC)',/,' ',
     &'36...LOCATION NAME __(LOCNAM)',/,'0',' ')
      CONTINUE
      END


      SUBROUTINE NEWDAT

C  ***************************************************************************
C  **  THIS SUBROUTINE RECEIVES AND PROCESSES DATA/INFORMATION    **
C  **  PERTAINING TO NEW INVENTORY ITEMS. FIRST ALL VALUES ARE    **
C  **  ASSIGNED ZEROS OR BLANKS, THEN THE USER IS PROMPTED FOR    **
C  **  EACH OF THE VALUES, AND THE GATHERED INFORMATION    THEN   **
C      WRITTEN INTO THE DATABASE..      **
C  ***************************************************************************

      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DAT2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR
      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,EXPTAV

      NAME='
      UNIT='
      MOB=' '
      CLASS=' '
      DEPCOD='
      CODLOC='
      LOCNAM='
      DAT1=0
      DATE2=0
      SOHAND=0.
      COST1=0.
      PLDTIM=0
      INVLEV=0
      DO 5 J=1,3
            ORDRNO(I)=0
            ORDRDT(I)=0
```

```
                ORDRQT(I)=0
                EXPTAV(I)=0
 5      CONTINUE

        DATE1=DATEA
        MM(2)=12

        ICALNO=8
        CALL INTROD

10      CALL PROMPT
        READ(*,20,ERR=10) AA
        SOHAND=AA
20      FORMAT(BN,F8.2)
        DO 30 I=13,19
          MM(2)=I
          CALL DALTER
30      CONTINUE

        DO 40 I=22,29
           MM(2)=I
           CALL DALTER
40      CONTINUE
        DO 50 I=35,36
           MM(2)=I
           CALL DALTER
50      CONTINUE

        OPEN(7,FILE='PARTNOS.INV',ACCESS='DIRECT',RECL=8)
        WRITE(7,'(A)',REC=IRECNO) PARTNO
        CONTINUE

        MM(2)=20
        CALL DWRITE

        MM(2)=30
        CALL DWRITE

        MM(2)=50
        CALL DWRITE

        CONTINUE
        WRITE(*,60)
60      FORMAT('0','IN THE NEXT PROMPT YOU CAN MAKE ANY ADDITIONAL',
       &' ALTERATIONS ',/,' ','THAT ARE NECESSARY WITH REGARD TO THIS',
       &' PARTICULAR ITEM')
        END


        SUBROUTINE DREAD

C  ***************************************************************************
C  **  THIS SUBROUTINE READS THE ITEM DATA/INFOR. FROM THE DATABASE **
C  ***************************************************************************
```

```
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)

      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM

      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &        EXPTAV,ORDRNO,ORDRDT,ORDRQT

      OPEN(4,FILE='DATA1.INV',ACCESS='DIRECT',RECL=72)
      OPEN(5,FILE='DATA2.INV',ACCESS='DIRECT',RECL=72)
      OPEN(6,FILE='DATA3.INV',ACCESS='DIRECT',RECL=72)

      READ(4,20,REC=IRECNO,END=50) PARTNO,NAME,UNIT,MOB,CLASS,
     &DEPCOD,DATE1,DATE2,CODLOC,LOCNAM
      READ(5,30,REC=IRECNO) SOHAND,HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,
     &ROLEV,PLDTIM,AVDEM,INVLEV
      READ(6,40,REC=IRECNO) ORDRNO(1),ORDRDT(1),EXPTAV(1),ORDRQT(1),
     &ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),ORDRDT(3),
     &EXPTAV(3),ORDRQT(3)

20    FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,1X,A8)
30    FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,1X,I8,
     &1X,I3)
40    FORMAT(I5,2(1X,I4),2(1X,I5),2(1X,I4),2(1X,I5),2(1X,I4),1X,I5)
      GO TO 70
50    CONTINUE
      MM(7)=IRECNO-1
70    CONTINUE
      END
      SUBROUTINE DWRITE

C  ***************************************************************************
C  **                                                                       **
C  **  THIS SUBROUTINE WRITES ITEM INFORMATION/STATUS INTO                  **
C  **  SYSTEM STORAGE. IT IS ONLY USED WHEN THE WRITING IS NOT              **
C  **  FREQUENT, OR ELSE THE USE OF THE SUBROUTINE WOULD BE                 **
C  **  AN INEFFICIENT PROCESS. WHEN THIS IS NOT SO THE EXECUTING            **
C  **  ROUTINE DOES THE WRITING ITSELF.                                     **
C  **                                                                       **
C  ***************************************************************************
```

```
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &            EXPTAV,ORDRNO,ORDRDT,ORDRQT

            OPEN(4,FILE='DATA1.INV', ACCESS='DIRECT', RECL=72)
            OPEN(5,FILE='DATA2.INV', ACCESS='DIRECT', RECL=72)
            OPEN(6,FILE='DATA3.INV', ACCESS='DIRECT', RECL=72)
       IF( MM(2).LE.23) THEN
           WRITE(5,20,REC=IRECNO) SOHAND,HOLCOS,COSTOD,COST1,SEVLEV,
     &            EOQTY,ROLEV,PLDTIM,AVDEM,INVLEV

        ELSEIF( MM(2).LE.40) THEN
                WRITE(4,30,REC=IRECNO) PARTNO,NAME,UNIT,MOB,CLASS,DEPCOD,
     &            DATE1,DATE2,CODLOC,LOCNAM

        ELSEIF( MM(2).LE.60) THEN
           WRITE(6,40,REC=IRECNO) ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &            ORDRQT(1),ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),
     &            ORDRNO(3),ORDRDT(3),EXPTAV(3),ORDRQT(3)
        ELSE
       ENDIF

20     FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,
     &1X,I8,1X,I3,6('='))
30     FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,
     &1X,A8,2('='))
40     FORMAT(I5,2(1X,I4),2(1X,I5),2(1X,I4),2(1X,I5),2(1X,I4),
     &1X,I5,7('='))
       END
```

```
      SUBROUTINE ORDERS

C  ***********************************************************************
C  **  THIS IS THE ORDER-PROCESSING SUBROUTINE. ORDER PLACEMENT **
C  **  AND ORDER RECEIPT IS HANDLED BY ACCESS TO THIS ROUTINE.  **
C  ***********************************************************************
C                    ****
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 USERCD
      CHARACTER*4 UNIT
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

      INTEGER EOQTY,ROLEV,PLDTIM,AVDEM,EXPTAV,ORDRNO,ORDRDT,ORDRQT,
     &        DATE1,DATE2,DATEA,DATEB

      ICALNO=6
         CALL INTROD
      IF(MODE.EQ.2) GOTO 75
10    WRITE(*,20)
20    FORMAT('0',' ENTER ONE[1] FOR PLACING AN ORDER, OR TWO[2] FOR',
     &' AN ORDER RECEIPT')
      WRITE(*,25)
25    FORMAT('0',' ENTER ANY OTHER DIGIT TO EXIT !')
      READ(*,'(I1)',ERR=10) II
        IF(II.GT.2) GOTO 500

      MM(2)=24
      MM(3)=1
      PAUSE 'NOTE: NO ORDER FOR A NEW INV.  [illegible] TAKEN !'
30    CALL PROMPT
      READ(*,'(A)',ERR=30) PARTNO

      CALL DATVER
      IF(ANS.EQ.NO) THEN
         PAUSE 'NON EXISTENT PART NO. ENTERED'
40    WRITE(*,50)
50    FORMAT('0','DO YOU WANT TO RE-ENTER PART NO ? Y/N')
      READ(*,60,ERR=40) ANS
60    FORMAT(A)
      IF(ANS.EQ.NO) GOTO 400
      IF(ANS.EQ.YES) GOTO 30
      ENDIF

      WRITE(*,70) IRECNO
70    FORMAT('0','INVENTORY RECORD NUMBER= ',I4,/,' ',30('-'))
```

```
      GOTO 79
75    II=2
79    CONTINUE

      IF( ORDRNO(3).NE.0) THEN
          MM(10)=3
        ELSEIF( ORDRNO(2).NE.0) THEN
          MM(10)=2
        ELSEIF( ORDRNO(1).NE.0) THEN
          MM(10)=1
        ELSE
          MM(10)=0
      ENDIF

      IF(II.EQ.2) THEN
80    WRITE(*,90) (ORDRNO(N),N=1,3)
90    FORMAT('0','THE FOLLOWING ORDERS ARE OUTSTANDING (0==>NONE)',
     &/,' ',36('-'),3I7)
100   WRITE(*,110) MM(10)
110   FORMAT('0','ENTER THE ORDER NUMBER[I5], OUT OF THE',I2,'THAT',
     &' IS BEING RECEIVED.',/,'0','NOTE THAT IF ORDER RECEIPT IS',
     &' PARTIAL YOU NEED TO RE-INPUT',/,' ',THE REMAINING ORDER QUAN',
     &'TITY USING SAME ORDER NUMBER !')
      READ(*,150,ERR=100) IORDER

      DO 114 N=1,3
        IF(IORDER.EQ.ORDRNO(N)) JJ=N
114   CONTINUE

      CALL ORDSLP(II,JJ)

      IF(JJ.EQ.1) THEN
          DO 115 N=1,2
             ORDRNO(N)=ORDRNO(N+1)
             ORDRDT(N)=ORDRDT(N+1)
             EXPTAV(N)=EXPTAV(N+1)
             ORDRQT(N)=ORDRQT(N+1)
115       CONTINUE
      ELSEIF(JJ.EQ.2) THEN
          ORDRNO(2)=ORDRNO(3)
          ORDRDT(2)=ORDRDT(3)
          ORDRQT(2)=ORDRQT(3)
          EXPTAV(2)=EXPTAV(3)
          CONTINUE
      ELSEIF(JJ.EQ.3) THEN
         CONTINUE
      ELSE
         WRITE(*,120)
120   FORMAT('0','***INVALID ORDER NUMBER****')
      GOTO 400
      ENDIF

      ORDRNO(3)=0
      ORDRDT(3)=0
```

```
        ORDRQT(3)=0
        EXPTAV(3)=0

        ELSEIF(II.EQ.1) THEN
               WRITE(*,90) (ORDRNO(N),N=1,3)
 130           WRITE(*,140)
 140           FORMAT('0','ENTER THE ORDER NUMBER FOR THE NEW ORDER',
     &' [I5]')
               READ(*,150,ERR=130) IORDER
 150           FORMAT(BN,I5)
 160           WRITE(*,170) IORDER
 170           FORMAT('0','ENTER THE QUANTITY-[I5] OF THE ORDER #',I6)
               READ(*,150,ERR=160) IQTTY
          JJ=MM(10)+1
          IF(JJ.LE.3) THEN
            ORDRNO(JJ)=IORDER
            ORDRDT(JJ)=DATEB
            ORDRQT(JJ)=IQTTY
          ELSE
            PAUSE 'MAXIMUM OF THREE[3] ORDERS PROCESSED !'
            GOTO 400
          ENDIF
 175     WRITE(*,180) PLDTIM
 180     FORMAT('0','THE PROCUREMENT LEAD TIME ON THIS ITEM IS ',I3,
     &' days.',/,' ','ENTER THE EXPECTED RECEIPT DATE__I4 [mo/dy]')
         READ(*,190,ERR=175) N
         EXPTAV(JJ)=N
 190     FORMAT(BN,I4)

         CALL ORDSLP(II,JJ)

         ENDIF
               MM(2)=50
               CALL DWRITE
 400     CONTINUE
 410     WRITE(*,420)
 420     FORMAT('0','MORE ORDERS TO BE PLACED/RECEIVED ? Y/N ?')
         READ(*,'(A)',ERR=410) ANS
         IF((ANS.NE.NO).AND.(MODE.EQ.1)) GOTO 10
 500     CONTINUE
         END
         SUBROUTINE DALTER

C  ***************************************************************
C  **                                                           **
C  **  THE SUBROUTINE, DALTER, IS USED TO ALTER THE INFORMA-    **
C  **  TION/DATA FOR AN ITEM FOLLOWING A TRANSACTION. IT IS     **
C  **  DESIGNED TO HANDLE THIS QUITE EASILY..                   **
C  **                                                           **
C  ***************************************************************
         COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
         COMMON /C2/ HOLCOS, COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &                AVDEM,INVLEV,SOHAND
         COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
```

```
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER EOQTY,ROLEV,PLDTIM,AVDEM

      MM(3)=1
      IF( MM(2).EQ.0) GOTO 220

      I=MM(2)
      IF(I.EQ.11) THEN
5           CALL PROMPT
            READ(*,10,ERR=5) AA
            SOHAND=SOHAND-AA
10          FORMAT(BN,F8.2)
       ELSEIF(I.EQ.12) THEN
15          CALL PROMPT
            READ(*,10,ERR=15) AA
            SOHAND=SOHAND+AA
            MODE=2
            CALL ORDERS
            MODE=1
      ELSEIF(I.EQ.13) THEN
18          CALL PROMPT
            READ(*,20,ERR=18) AA
            HOLCOS=AA
20          FORMAT(BN,F6.2)
      ELSEIF(I.EQ.14) THEN
25          CALL PROMPT
            READ(*,20,ERR=25) AA
            COSTOD=AA
      ELSEIF(I.EQ.15) THEN
28          CALL PROMPT
            READ(*,10,ERR=28) AA
            COST1=AA
      ELSEIF(I.EQ.16) THEN
30          CALL PROMPT
            READ(*,33,ERR=30) AA
            SEVLEV=AA
33          FORMAT(BN,F4.2)
      ELSEIF(I.EQ.17) THEN
36          CALL PROMPT
            READ(*,40,ERR=36) EOQTY
40          FORMAT(BN,I5)
      ELSEIF(I.EQ.18) THEN
44          CALL PROMPT
            READ(*,40,ERR=44) ROLEV
      ELSEIF(I.EQ.19) THEN
```

```
48          CALL PROMPT
            READ(*,50,ERR=48) PLDTIM
50          FORMAT(BN,I4)
      ELSEIF(I.EQ.22) THEN
55          CALL PROMPT
            READ(*,56,ERR=55) AVDEM
56          FORMAT(BN,I8)
      ELSEIF(I.EQ.23) THEN
58          CALL PROMPT
            READ(*,60,ERR=58) INVLEV
60          FORMAT(BN,I3)
      ELSEIF(I.EQ.24) THEN
61          WRITE(*,62)
62          FORMAT('0','ARE YOU SURE YOU WANT TO CHANGE PART #? ')
            READ(*,'(A)',ERR=61) ANS
            IF(ANS.NE.YES) GO TO 68

66          CALL PROMPT
            READ(*,70,ERR=66) PARTNO
68          CONTINUE
70          FORMAT(A)
      ELSEIF(I.EQ.25) THEN
80          CALL PROMPT
            READ(*,90,ERR=80) NAME
90          FORMAT(A)
      ELSEIF(I.EQ.26) THEN
100         CALL PROMPT
            READ(*,110,ERR=100) UNIT
110         FORMAT(A)
      ELSEIF(I.EQ.27) THEN
120         CALL PROMPT
            READ(*,110,ERR=120) MOB
      ELSEIF(I.EQ.28) THEN
140         CALL PROMPT
            READ(*,110,ERR=140) CLASS
      ELSEIF(I.EQ.29) THEN
150         CALL PROMPT
            READ(*,110,ERR=150) DEPCOD
      ELSEIF(I.EQ.35) THEN
170         CALL PROMPT
            READ(*,110,ERR=170) CODLOC
      ELSEIF(I.EQ.36) THEN
190         CALL PROMPT
            READ(*,110,ERR=190) LOCNAM

      ELSE
            PAUSE '**INVALID PARAMETER CODE !'
      ENDIF

220   CONTINUE
      END
```

```
      SUBROUTINE ENQURY
C     ***********************************************************
C     **        THIS IS THE ENQUIRY SUBROUTINE.                **
C     **  IT IS USED TO PROVIDE ITEM DATA/INFORMATION AS REQUIRED **
C     **                    BY THE USER..                      **
C     **                      **
C     ***********************************************************

      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,BLK,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,EXPTAV,
     &            AVDEM,ORDRNO,ORDRDT,ORDRQT

      BLK='        '
      ICALNO=5
        CALL INTROD
10    CONTINUE

      MM(3)=2
      MM(2)=24
12    CONTINUE
      WRITE(*,15)
15    FORMAT('0','ENTER BLANKS [8] FOR PART NO. TO EXIT !')
20    CALL PROMPT
      READ(*,30,ERR=20) PARTNO
30    FORMAT(A)
      IF(PARTNO.EQ.BLK) GO TO 500

      CALL DATVER
      YES='P'
      IF(ANS.NE.YES) GOTO 400
      YES='Y'
40    WRITE(*,50) PARTNO
50    FORMAT('0','ENTER ONE [1] IF YOU WANT DETAIL INFORMATION ON',
     &' THE PART NO. ',A,/,' ','OR ENTER ANY OTHER DIGIT IF ONLY',
     &' PARTIAL INFORMATION IS REQUIRED.')
      READ(*,'(I1)',ERR=40) I
      PAUSE 'CONTROL PrtSc IF PRINTER OUTPUT IS NEEDED !'
```

```
      IF(I.NE.1) THEN
      WRITE(*,60) PARTNO,NAME
60    FORMAT('0','PART NUMBER:',3X,A,/,13X,11('='),/,'0',
     &'PART DESCRIPTION:',2X,A)
      WRITE(*,70) SOHAND,COST1,HOLCOS,COSTOD,EOQTY,ROLEV,PLDTIM
70    FORMAT('0',' N  HAND ',' UNIT COST',' HOL COST ',' ORD COST ',
     &'EC.ORD.QTY',' RE ORD LE ','PROC/LTIME',/,2X,8('*'),2X,9('*'),
     &1X,8('*'),2X,8('*'),1X,10('*'),1X,9('*'),1X,10('*'),/,' ',
     &4(F10.2),3(I10))

75    WRITE(*,80) PARTNO
80    FORMAT('0','DO YOU NEED FURTHER INFORMATION ON ITEM # ',A,' ?')
      READ(*,90,ERR=75) ANS
90    FORMAT(A)
      IF(ANS.EQ.NO) GO TO 400
      WRITE(*,100) AVDEM,SEVLEV,INVLEV,MOB,CLASS,UNIT,DEPCOD
100   FORMAT('0',' AVER DEM',' SERV LEV',' INV LEV',' M OR BUY',
     &' INV CLASS',' MEAS UNIT',' DEPT COD',/,' ',2X,8('*'),2X,8('*'),
     &3X,7('*'),2X,8('*'),1X,9('*'),1X,9('*'),2X,8('*'),/,' ',I10,
     &F10.2,I10,9X,A,9X,A,6X,A,5X,A)

110   WRITE(*,80) PARTNO
      READ(*,90,ERR=110) ANS
      IF(ANS.EQ.NO) GOTO 400
      WRITE(*,120) CODLOC,LOCNAM,ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &ORDRQT(1)
120   FORMAT('0',' LOCN CODE',' LOCN NAME',' ORD NO 1:',' ORD.1 DAT',
     &' EXP. ARR ',' ORD.1 QTY',/,' ',1X,9('*'),1X,9('*'),1X,9('*'),
     &1X,9('*'),1X,8('*'),2X,9('*'),/,' ',7X,A,3X,A,2X,I7,3(I10))
130   WRITE(*,80) PARTNO
      READ(*,90,ERR=130) ANS
      IF(ANS.EQ.NO) GOTO 400
      WRITE(*,140)ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),
     &ORDRDT(3),EXPTAV(3),ORDRQT(3)
140   FORMAT('0',' ORD # 2',' DATE:',' EXP.ARR',' QUATTY',
     &' ORD # 3',' DATE:',' EXP.ARR',' QUATTY',/,' ',1X,7('*'),3X,
     &5('*'),1X,7('*'),2X,6('*'),1X,7('*'),3X,5('*'),1X,7('*'),2X,
     &6('*'),/,' ',8(I8))

      ELSE
         WRITE(*,60) PARTNO,NAME
         WRITE(*,70) SOHAND,COST1,HOLCOS,COSTOD,EOQTY,ROLEV,PLDTIM
         WRITE(*,100) AVDEM,SEVLEV,INVLEV,MOB,CLASS,UNIT,DEPCOD
         WRITE(*,120) CODLOC,LOCNAM,ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &ORDRQT(1)
         WRITE(*,140) ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),
     &ORDRNO(3),ORDRDT(3),EXPTAV(3),ORDRQT(3)

      ENDIF
400   CONTINUE
      YES='Y'
      CONTINUE
410   WRITE(*,420)
420   FORMAT('0','IS INFORMATION NEEDED FOR (AN)OTHER ITEM(S) ?')
```

```
      READ(*,90,ERR=410) ANS
      IF(ANS.EQ.YES) GOTO 12
500   CONTINUE
      END
      SUBROUTINE EXCEPT

C     ************************************************************
C     *                                                          *
C     *  THIS SUBROUTINE IS USED TO GENERATE EXCEPTION REPORTS   *
C     *  WHENEVER SOMETHING REQUIRING PROMPT ACTION IS DETECTED  *
C     *  DURING A COMPLETE REVIEW OF THE DATA-BASE..             *
C     *                                                          *
C     ************************************************************
C
      COMMON /C1/ ICALNO, MM(10), MODE, IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD,BLK,CDN,COD(10)
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR

      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &            EXPTAV,ORDRNO,ORDRDT,ORDRQT

      ICALNO=2
      CALL INTROD

      BLK='  '
      CDN='LA'
        IRECNO=0
        J=0
      CONTINUE
      PAUSE '!!!!!!!! MAKE SURE PRINTER IS ON..!!!!!!!'
      WRITE(*,8)
8     FORMAT('0','.....HOLD ON ! EXCEPTION REPORT IN PROCESS...')
      OPEN(2,FILE='PRN')
      OPEN(4,FILE='DATA1.INV',ACCESS='DIRECT',RECL=72)
      OPEN(5,FILE='DATA2.INV',ACCESS='DIRECT',RECL=72)
      OPEN(6,FILE='DATA3.INV',ACCESS='DIRECT',RECL=72)

10    IRECNO=IRECNO+1
      DO 20 I=1,10
        COD(I)=BLK
20    CONTINUE
```

```
      READ(4,25,REC=IRECNO,END=40) PARTNO,NAME,UNIT,MOB,CLASS,
     &DEPCOD,DATE1,DATE2,CODLOC,LOCNAM
      READ(5,30,REC=IRECNO,END=40) SOHAND,HOLCOS,COSTOD,COST1,
     &SEVLEV,EOQTY,ROLEV,PLDTIM,AVDEM,INVLEV
      READ(6,35,REC=IRECNO) ORDRNO(1),ORDRDT(1),EXPTAV(1),ORDRQT(1),
     &ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),ORDRDT(3),
     &EXPTAV(3),ORDRQT(3)

25    FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,1X,A8)
30    FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,1X,I8,
     &1X,I3)
35    FORMAT(I5,2(1X,I4),2(1X,I5),2(1X,I4),2(1X,I5),2(1X,I4),1X,I5)

      GO TO 45
40    CONTINUE
      MM(7)=IRECNO-1
45    CONTINUE

         IF( MM(7).NE.0) GOTO 150
      IF(SOHAND.LE.ROLEV) COD(1)='SL'
      IF((EXPTAV(1).LT.DATEB).AND.(EXPTAV(1).NE.0)) COD(2)=CDN
      IF((EXPTAV(2).LT.DATEB).AND.(EXPTAV(2).NE.0)) COD(3)=CDN
      IF((EXPTAV(3).LT.DATEB).AND.(EXPTAV(3).NE.0)) COD(4)=CDN
      IF(COD(1).NE.BLK) THEN
            GOTO 50
      ELSEIF(COD(2).EQ.CDN) THEN
            GOTO 50
      ELSEIF(COD(3).EQ.CDN) THEN
            GOTO 50
      ELSEIF(COD(4).EQ.CDN) THEN
            GOTO 50
      ENDIF
      GOTO 100
50    CONTINUE
      IF(J.EQ.0) THEN
            CALL THEAD
            PAR='**E X C E P T I O N    R E P O R T**'
            CALL RTHEAD
            WRITE(2,60)
60    FORMAT('0',' ',/,' ','ITEM NUMBER:',2X,'CODE1',2X,'CODE2',2X,
     &'CODE3',2X,'CODE4',2X,'CODE5',2X,'CODE6',2X,'CODE7',2X,'CODE8',
     &/,70('-'))

         J=J+1
      ELSE
      ENDIF
      WRITE(2,70) PARTNO, (COD(I),I=1,8)
70    FORMAT(5X,A,8(5X,A))

100   CONTINUE
         GOTO 10
150   CONTINUE
      MM(7)=0
      IF(J.GT.0) THEN
```

```
      READ(4,25,REC=IRECNO,END=40) PARTNO,NAME,UNIT,MOB,CLASS,
     &DEPCOD,DATE1,DATE2,CODLOC,LOCNAM
      READ(5,30,REC=IRECNO,END=40) SOHAND,HOLCOS,COSTOD,COST1,
     &SEVLEV,EOQTY,ROLEV,PLDTIM,AVDEM,INVLEV
      READ(6,35,REC=IRECNO) ORDRNO(1),ORDRDT(1),EXPTAV(1),ORDRQT(1),
     &ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),ORDRDT(3),
     &EXPTAV(3),ORDRQT(3)

25    FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,1X,A8)
30    FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,1X,I8,
     &1X,I3)
35    FORMAT(I5,2(1X,I4),2(1X,I5),2(1X,I4),2(1X,I5),2(1X,I4),1X,I5)

      GO TO 45
40    CONTINUE
      MM(7)=IRECNO-1
45    CONTINUE

         IF( MM(7).NE.0) GOTO 150
      IF(SOHAND.LE.ROLEV) COD(1)='SL'
      IF((EXPTAV(1).LT.DATEB).AND.(EXPTAV(1).NE.0)) COD(2)=CDN
      IF((EXPTAV(2).LT.DATEB).AND.(EXPTAV(2).NE.0)) COD(3)=CDN
      IF((EXPTAV(3).LT.DATEB).AND.(EXPTAV(3).NE.0)) COD(4)=CDN
      IF(COD(1).NE.BLK) THEN
            GOTO 50
      ELSEIF(COD(2).EQ.CDN) THEN
            GOTO 50
      ELSEIF(COD(3).EQ.CDN) THEN
            GOTO 50
      ELSEIF(COD(4).EQ.CDN) THEN
            GOTO 50
      ENDIF
      GOTO 100
50    CONTINUE
      IF(J.EQ.0) THEN
            CALL THEAD
            PAR='**E X C E P T I O N    R E P O R T**'
            CALL RTHEAD
            WRITE(2,60)
60    FORMAT('0',' ',/,' ','ITEM NUMBER:',2X,'CODE1',2X,'CODE2',2X,
     &'CODE3',2X,'CODE4',2X,'CODE5',2X,'CODE6',2X,'CODE7',2X,'CODE8',
     &/,70('-'))

         J=J+1
      ELSE
      ENDIF
      WRITE(2,70) PARTNO, (COD(I),I=1,8)
70    FORMAT(5X,A,8(5X,A))

100   CONTINUE
         GOTO 10
150   CONTINUE
      MM(7)=0
      IF(J.GT.0) THEN
```

```
        WRITE(2,155)
155     FORMAT('0','  CODE KEY:',/,12('='),/,'0','  SL....',
     &'STOCK LEVEL AT/BELOW ROLEV.',/,'0','  LA....LATE ARRIVAL',
     &' OF ORDER [may require expediting!]',/,5X,'CODE2-->ORDER1',
     &/,5X,'CODE3-->ORDER2',/,5X,'CODE4-->ORDER3',/)
        ELSE
          WRITE(*,158)
158     FORMAT('0','....NO EMERGENCY !......NO REPORT',/,'0',' ')
        ENDIF

        WRITE(2,160)
160        FORMAT('0',25('*'),' END ',25('*'))
        END


        SUBROUTINE CHGREP

C       ***************************************************************
C       *                    *
C       *  THIS SUBROUTINE GENERATES CHANGE REPORTS FOLLOWING   *
C       *  REGULAR TRANSACTIONS TO INDICATE THE NEW STATUS OF   *
C       *  ITEMS FOR WHICH TRANSACTIONS WERE MADE..             *
C       *                    *
C       ***************************************************************

        COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
        COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &              AVDEM,INVLEV,SOHAND
        COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
        COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
        COMMON /C5/ DATE1,DATE2,DATEA,DATEB
        COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
        COMMON /C7/ ITEREC(100),NTOTAL
        CHARACTER*1 ANS,NO,YES,MOB,CLASS
        CHARACTER*2 CODLOC,USERCD,BLK
        CHARACTER*4 UNIT
        CHARACTER*5 DEPCOD
        CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
        CHARACTER*20 NAME
        CHARACTER*36 PAR
        INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &          EXPTAV,ORDRNO,ORDRDT,ORDRQT


        IF(NTOTAL.EQ.0) THEN
          PAUSE '** ZERO TRANSACTIONS ENTERED ** NO REPORT !'
          GOTO 200
        ENDIF
          PAR='**T R A N S A C T I O N S    R P T**'
          CALL THEAD
          CALL RTHEAD
        OPEN(2,FILE='PRN')
        OPEN(4,FILE='DATA1.INV',ACCESS='DIRECT',RECL=72)
        OPEN(5,FILE='DATA2.INV',ACCESS='DIRECT',RECL=72)
```

```
      OPEN(6,FILE='DATA5.INV',ACCESS='DIRECT',RECL=72)

      WRITE(2,10) NTOTAL
10    FORMAT('0',20X,'TOTAL TRANSACTIONS = ',I3,/,' ',20X,23('='))
      WRITE(2,20)
20    FORMAT('0','TRANS.#',2X,'ITEM.NUMBER:',2X,'NEW O/HAND',2X,
     &'ROLEV',2X,'OUTSTANDING ORDER Q/TITIES')
      WRITE(2,30)
30    FORMAT(' ',7('='),2X,12('='),2X,10('='),2X,5('='),2X,26('='))
      CONTINUE

      DO 100 I=1, NTOTAL
        IRECNO=ITEREC(I)

      READ(4,40,REC=IRECNO,END=70) PARTNO,NAME,UNIT,MOB,CLASS,
     &DEPCOD,DATE1,DATE2,CODLOC,LOCNAM
      READ(5,50,REC=IRECNO,END=70) SOHAND,HOLCOS,COSTOD,COST1,
     &SEVLEV,EOQTY,ROLEV,PLDTIM,AVDEM,INVLEV
      READ(6,60,REC=IRECNO,END=70) ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &ORDRQT(1),ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),
     &ORDRDT(3),EXPTAV(3),ORDRQT(3)
40    FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,1X,A8)
50    FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,1X,I8,
     &1X,I3)
60    FORMAT(I5,2(1X,I4),2(1X,I5),2(1X,I4),2(1X,I5),2(1X,I4),1X,I5)
        WRITE(2,65) I,PARTNO,SOHAND,ROLEV,(ORDRQT(IX),IX=1,3)
65    FORMAT(' ',4X,I3,6X,A,4X,F8:2,3X,I4,3(5X,I4))
100   CONTINUE
70    CONTINUE
      WRITE(2,110)
110   FORMAT('0',25('*'),' END ',25('*'))
200   CONTINUE
      END
$SUBTITLE:'SORTING OF REQUIRED ITEM TYPES FROM DATA'
C
C ***************************************************************************
C **       THIS IS THE SORTING SUBROUTINE. ITEMS OF A GIVEN      **
C ** SIGNIFICANT CODE PATTERN  ARE SEARCHED AND OUTPUT EITHER **
C ** ON THE PRINTER OR ON THE SCREEN AS REQUIRED.                **
C **                              **.
C ***************************************************************************

      SUBROUTINE SORTIT

      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
      COMMON /C7/ ITEREC(100),NTOTAL
      COMMON /C8/ MATRL(100)

      CHARACTER*1 ANS,ENT,FF(4),HH(4),PP,BLK,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*4 CLASSN,IEND
      CHARACTER*8 PARTNO,DATE,INIT,MATRL,X
      CHARACTER*36 PAR
```

```
C
15    WRITE(*,20)
20    FORMAT('0','THIS PROGRAM HANDLES THE FIRST FOUR CHARACTERS OF THE
     +ITEMCODES.',/,' ','A SEARCH IS MADE FOR ITEMS/PARTS WITH THE',
     +'SPECIFIED ID. CODES',/,' ','IN THE GIVEN POSITIONS. YOU WILL',
     +' BE TOLD OF THE TOTAL NO.',/,' ','OF ITEMS IN QUESTION AND',
     +' ASKED WHETHER THEY SHOULD BE PRINTED',/,' ','ON THE PRINTER',
     +'OR ON THE SCREEN',/,'0','**INPUT THE REQUIRED CHARACTERS--',
     &'LEAVING BLANKS ELSEWHERE .. eg. D_2_')
      READ(*,40) HH(1),HH(2),HH(3),HH(4)
40    FORMAT(4A1)
      OPEN(7,FILE='PARTNOS.INV', ACCESS='DIRECT', RECL=8)
C
C
      PP=' '
      I=1
      J=0
50    J=J+1

      READ(7,55,REC=J,END=70) FF(1),FF(2),FF(3),FF(4),IEND
C
55    FORMAT(4A1,A)
      IF(HH(1).EQ.PP) THEN
          CONTINUE
      ELSE
          IF(HH(1).NE.FF(1)) GO TO 50
      ENDIF
C
      IF(HH(2).EQ.PP) THEN
          CONTINUE
      ELSE
          IF(HH(2).NE.FF(2)) GO TO 50
      ENDIF
C
         IF(HH(3).EQ.PP) THEN
             CONTINUE
         ELSE
             IF(HH(3).NE.FF(3)) GOTO 50
         ENDIF
C
      IF(HH(4).EQ.PP) THEN
             CONTINUE
      ELSE
             IF(HH(4).NE.FF(4)) GO TO 50
      ENDIF
      CONTINUE
C
      ITEREC(I)=J
      WRITE(X,55) FF(1),FF(2),FF(3),FF(4),IEND
      READ(X,'(A8)') MATRL(I)

         I=I+1
      GO TO 50
70    CONTINUE
```

```
C
C
        NTOTAL=I-1
      IF(NTOTAL.EQ.0) THEN
        PAUSE ' TOTAL NO. APPLICABLE=00 (NIL !!) '
        GO TO 100
      ENDIF

        WRITE(*,80) NTOTAL
80    FORMAT('0',12X,'TOTAL APPLICABLE #. OF ITEMS=',I4,//,
     +12X,'DO YOU WANT THEM ON PRINTER?',/,15X,
     +'ENTER Y FOR YES OR',/,15X,'ENTER N FOR SCREEN O/P OR',/,15X,
     +'ENTER A BLANK FOR NO OUTPUT AT ALL..')
C

      READ(*,'(A1)',ERR=100) ANS
      ENT='Y'
      BLK=' '
      IF(ENT.EQ.ANS) THEN
         OPEN(1,FILE='PRN')
       PAR='   MATERIAL  PARTNO  SORT  PRINTOUT '
       MODE=2
       CALL RTHEAD
       MODE=1
      ELSEIF(ANS.EQ.BLK) THEN
          GO TO 100
      ELSE
         OPEN(1,FILE='USER')
      ENDIF
C
      WRITE(1,140)
      WRITE(1,120)
C
      DO 105 M=1, NTOTAL
          WRITE(1,130) M,MATRL(M)
C
120   FORMAT('0',8X,'SER.#',4X,'ITEM ID. NO.',4X,'DESCRIPTION',/,
     &' ',8X,5('='),4X,12('='),4X,11('='))
130   FORMAT(8X,I4,7X,A8)
140   FORMAT('0',8X,'_____',//)
      WRITE(1,140)
C
105   CONTINUE
100   CONTINUE
C
      WRITE(*,150)
150   FORMAT('0',10X,'DO YOU WANT ANOTHER SORTING ? ..Y OR N ?')
      READ(*,'(A1)') ANS
      ENT='Y'
      IF(ENT.EQ.ANS) GO TO 15
      RETURN
      END
```

```
$SUBTITLE:'ITEM IDENTIFICATION/ DATA PROC. PROGRAM'
C
C ***********************************************************************
C                                **
C  THIS PROGRAM IS INTENDED TO ASSIST IN THE DEVELOPMENT          **
C  OF NEW ITEM CODES ACCORDING TO THE CRETERIA OUTLINED UNDER     **
C  'INVENTORY IDENTIFICATION', AS WELL AS FOR USE IN THE          **
C  EVALUATION (DATA PROCESSING) OF VARIOUS PARAMETERS:            **
C                                **
C ***********************************************************************

      PROGRAM MAIN33

      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
        CHARACTER*1 COD(4),ENT,ANS,BLK,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

      CALL ENTRES
      IF( MM(4).EQ.3) GOTO 560

      WRITE(*,2)
2     FORMAT(' ',60('*'),/,' ','**',/,' ','**  ','THIS IS THE',
     &' INVENTORY IDENTIFICATION PROGRAM. IT IS AN AID',/,' ','**  ',
     &'IN THE ASSIGNMENT OF PART IDS. AS AN OPTION',/,' ','**  ',
     &'SORTING (SUBROUTINE SORT) CAN BE DONE FOR OTHER ITEMS WITH',/,
     &' ','**  ','SIMILAR CHARACTERS TO AVOID ANY ID. DUPLICATION..',
     &/,' ','**  ',/,' ',60('*'))

      WRITE(*,4)
4     FORMAT('0','FOLLOW THE FOLLOWING PROMPTS CAREFULLY TO',
     &' DETERMINE WHICH',/,' ','SIGNIFICANT CODES SHOULD BE ASSIGNED',
     &' TO THE ITEM IN QUESTION:')

      PAUSE 'THE SERIES OF PROMPTS FOLLOW FROM HERE !!'

5     WRITE(*,10)
10    FORMAT(10X,'FINAL PRODUCT /SPECIAL INVENTORY CLASS CODE',/,
     +10X,'-----------------------------------------',/)
      BLK=' '
      ENT='Y'
      WRITE(*,20)
20    FORMAT(11X,'WOULD YOU CONSIDER THE ITEM IN QUESTION TO',/,10X,
     +'BELONG TO ONE OF THE FOLLOWING SPECIAL CLASSES?',/,10X,
     +'IF SO ENTER THE APPROPRIATE LETTER - ELSE THEN',/,10X,
     +'ENTER A BLANK',/,15X,'E: ELECTRICAL',/,15X,
     +'H: HYDRAULICS AND PNEUMATICS',/,15X,'T: TOOLS',/,15X
     +,'X: STANDARD ITEMS/COMMERCIALS (eg.BOLTS & NUTS)',/,15X,
     +'ENTER A BLANK FOR -NONE OF THE ABOVE:')
      READ(*,'(A1)',ERR=40) ANS
        IF(ANS.EQ.BLK) GO TO 40
```

```
          COD(1)=ANS
        CONTINUE
       GO TO 110
40     WRITE(*,50)
50     FORMAT('0','WOULD YO  CONSIDER THE ITEM TO BE PARTICULARLY',/,
      +' ','USED FOR ONE OF THE COMPANY''S PRODUCTS AS OUTLINED BELOW?',
      +/,' ','THIS APPLIES ALSO FOR AN ITEM THAT IS MAINLY USED',/,' ',
      +'FOR A PARTICULA  PRODUCT BUT ALSO USED ..TO A LESS',/,' ',
      +'EXTENT IN ANOTHER/OTHERS.',/)
       WRITE(*,60)
60     FORMAT(16X,'THE PRODUCTS LIST NOW FOLLOWS',/)
       PAUSE
       WRITE(*,70)
70     FORMAT(12X,'A: ELEVATING DOCKS',/,12X,'B: DOCK BOARDS /PLATES',
      +/,12X,'C: CONVEYORS',/,12X,'D: DOCK LEVELLERS',/,12X,
      +'F: FORKLIFTS AND ATTACHMENTS & PALLET TRUCKS',/,12X,
      +'K: CRANES AND MONORAILS',/,12X,'L: LADDERS',/,12X,
      +'N: HAND TRUCKS',/,12X,'P: PLATFORM TRUCKS & TRUCKS GENERAL',
      +/,12X,'R: RACKING AND SHELVING',/,15X,
      +'IF NONE ENTER A BLANK!',/)
C
       READ(*,'(A1)',ERR=80) ANS
         IF(ANS.EQ.BLK) GO TO 80
        COD(1)=ANS
       CONTINUE
       GOTO 110
80     WRITE(*,90)
90     FORMAT(12X,'THE ITEM SHOULD PROBABLY BE CLASSIFIED UNDER',/,12X,
      +'*SPECIAL EQUIPT/ MATERIAL NOT IN ABOVE GROUPS',/,12X,
      +'WOULD YOU LIKE TO GO THROUGH THE ABOVE CHECK AGAIN?',/,12X,
      +'IF SO ENTER Y',4X,'ELSE ENTER N',/)
       READ(*,'(A1)',ERR=100) ANS
       IF(ANS.EQ.ENT) THEN
          GO TO 5
       ELSE
          CONTINUE
       ENDIF

100    COD(1)='S'
110    WRITE(*,120) COD(1)
120    FORMAT(20X,'PRODUCT CODE=',3X,A1,/,20X,'-----------------',/)
140    WRITE(*,150)
150    FORMAT('0',10X,'INVENTORY LEVEL AND PRODUCTION PROCESS CODE')
       WRITE(*,130)
130    FORMAT(' ',12X,'**************************************')
       WRITE(*,160)
160    FORMAT('0',6X,'IF AN ITEM HAS CHARACTERISTICS OF MORE THAN',/,5X,
      +'ONE CODE, THE PREDOMINANT ONE SHOULD BE PICKED.',/,5X,
      +'eg. IF A PART IS BOTH FORMED AND WELDED, THE MORE APPARENT',/,
      +5X,'PROCESS SAY, IF IT IS WELDING THEN IT SHOULD BE CODED',/,5X,
      +'ACCORDINGLY.'/)
       WRITE(*,165)
165    FORMAT(/)
       WRITE(*,170)
```

```
170    FORMAT(5X,'IN EACH CASE ENTER ONE OF THE GIVEN CODES ONLY IF',/,
      +5X,'APPLICABLE, OTHERWISE JUST ENTER A BLANK',5X,'YOU WILL BE',/,
      +5X,'GIVEN A CHANCE TO GO THROUGH THIS AGAIN IF AT THE END',/,5X,
      +'NO CODE IS CHOSEN')
       PAUSE 'THE SEQUENCE NOW FOLLOWS'
       WRITE(*,180)
180    FORMAT('0','IS THE ITEM ONE OF THE FOLLOWING?',/,10X,12('-'))
       WRITE(*,190)
190    FORMAT('0',10X,'9: A FINISHED PRODUCT?',/,10X,
      +'8: A MAJOR/FINAL ASSEMBLY?',/,10X,'7: A SUB-ASSEMBLY?',/,'0',
      +10X,'IF NONE ENTER A BLANK')
       READ(*,'(A1)',ERR=200) ANS
           IF(ANS.EQ.BLK) GO TO 200
          COD(2)=ANS
         CONTINUE
       GOTO 270
200    WRITE(*,180)
       WRITE(*,210)
210    FORMAT('0',9X,'0: RAW MATERIAL, UNPROCESSED?',/,10X,
      +'1: RAW MATERIAL, SINGLE ITEM?',/,10X,
      +'2: RAW MATERIAL, OTHER THAN ABOVE?',/,5X,'UNDERGONE NO',1X,
      +'PROCESSING IN THE COMPANY AT ALL!',/,10X,
      +'IF NONE ENTER A BLANK')
       READ(*,'(A1)',ERR=220) ANS
           IF(ANS.EQ.BLK) GO TO 220
          COD(2)=ANS
         CONTINUE
         GOTO 270
220    WRITE(*,180)
       WRITE(*,230)
230    FORMAT(10X,'3: SINGLE COMPONENT, CAST OR MACHINED',/,10X,
      +'4: SINGLE COMPONENT, FORGED OR FORMED',/,10X,
      +'5: INDIVIDUAL ITEM, WELDED',/,10X,
      +'6: INDIVIDUAL ITEM, BY OTHER PROCESS/ OPERATION',/,15X,
      +'IF NONE OF THESE ENTER A BLANK')
       READ(*,'(A1)',ERR=240) ANS
         IF(ANS.EQ.BLK) GO TO 240
        COD(2)=ANS
         CONTINUE
         GOTO 270
240    WRITE(*,250)
250    FORMAT(5X,'ONE OF THE TEN(10) CATEGORIES SHOULD APPLY.',/,5X,
      +'ANOTHER LOOK WILL HELP!',/)
       GO TO 140
270    CONTINUE
       WRITE(*,280) COD(2)
280    FORMAT(10X,'INVENTORY LEVEL AND PRODN. PROCESS CODE=',3X,A1)
       WRITE(*,130)
290    WRITE(*,300)
300    FORMAT(/)
310    WRITE(*,320)
320    FORMAT('0',15X,'GEOMETRIC FORM OF INVENTORY CODE')
       WRITE(*,130)
       WRITE(*,330)
```

```
330   FORMAT('0',5X,'THE GEOMETRIC CODE MAINLY APPLIES TO SINGLE ITEMS',
     +/,5X,'AND SIMPLE SUB-ASSEMBLIES/ASSEMBLIES, OTHERWISE CODE',/,
     +5X,'#7 FOR SOLIDS AND CODE #9 FOR FLUIDS WILL APPLY.',/,
     +5X,'WHERE BOTH 3 & 4 APPLY TAKE 4.',/,5X,
     +'GO CAREFULLY THRU THE LIST & ENTER THE RIGHT CODE.',/)
      WRITE(*,340)
340   FORMAT(11X,'0: PLATE OR SHEET MATERIAL',/,10X,
     +'1: NON- CYLINDRICAL AND HOLLOW',/,10X,
     +'2: NON- CYLINDRICAL, OTHER THAN 1',/,10X
     +'3: CYLINDRICAL HOLLOW',/,10X,'4: CYLINDRICAL THREADED',/,10X,
     +'5: CYLINDRICAL OTHER THAN 3 OR/AND 4',/,10X,'6: ---',/,10X,
     +'7: SOLID. NONE OF THE ABOVE (NON- SIMPLE ITEMS ETC.)',/,10X,
     +'8: ---',/,10X,'9: FLUID',/,8X,'---------')
      READ(*,'(A1)',ERR=350) COD(3)
        CONTINUE
        GO TO 385
350   WRITE(*,360)
360   FORMAT(10X,'ENTRY ERROR - NO ENTRY MADE')
      WRITE(*,370)
370   FORMAT(10X,'DO YOU WANT TO CONTINUE? Y / N ')
      READ(*,'(A1)',ERR=350) ANS
      IF(ANS.EQ.ENT) GO TO 380
         GOTO 530
380   GOTO 310
385   WRITE(*,390) COD(3)
390   FORMAT(10X,'GEOMETRIC FORM CODE =',3X,A1,/,10X,'_____________
     +_____',/)
      WRITE(*,400)
400   FORMAT('0',15X,'DRAWING  CODE')
      WRITE(*,130)
405   WRITE(*,410)
410   FORMAT('0',5X,'ANSWER THE FOLLOWING QUESTIONS WITH REGARD TO',/,
     +10X,'THE ITEMS DRAWINGS, IF ANY, CAREFULLY***')
       WRITE(*,420)
420   FORMAT(11X,'IS THE ITEM EXPECTED TO HAVE A DRAWING?',/,10X,
     +'(EITHER NOW OR LATER!) ...Y OR N? ENTER ONE')
      READ(*,'(A1)',ERR=430) ANS
        IF(ANS.EQ.ENT) THEN
         CONTINUE
        ELSE
         COD(4)='N'
         GOTO 480
        ENDIF
      GOTO 450
430   WRITE(*,360)
      WRITE(*,440)
440   FORMAT(10X,'DO YOU WANT TO CONTINUE? Y / N?')
      READ(*,'(A1)',ERR=430) ANS
       IF(ANS.EQ.ENT) GO TO 405
       GOTO 530
450   WRITE(*,460)
460   FORMAT(5X,'IS A DRAWING CURRENTLY AVAILABLE (A) OR IS IT',/,5X,
     +'PENDING (P)? ENTER  A or P')
      READ(*,'(A1)',ERR=430) ANS
```

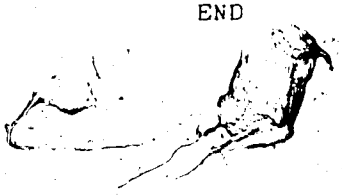
```
      ENT='P'
      IF(ANS.EQ.ENT) THEN
        COD(4)=ANS
        ENT='Y'
        GOTO 480
      ELSE
        ENT='Y'
      ENDIF
      WRITE(*,470)
470   FORMAT(5X,'INDICATE THE APPLICABLE DRAWING SIZE',/,8X,
     +'A or    B or      C or      D or     E.',/,8X,
     +'smallest          >>>>>>      BIGGEST',/,2X,
     +'enter one of A/B/C/D/E')
      READ(*,'(A1)',ERR=430) ANS
        COD(4)=ANS
480   CONTINUE
      WRITE(*,490) COD(4)
490   FORMAT(10X,'DRAWING CODE =',2X,A1,/,10X,'____________________')
      WRITE(*,300)
      WRITE(*,500) COD(1), COD(2), COD(3), COD(4)
500   FORMAT('0',5X,'THE FIRST FOUR ITEM CODES SHOULD BE:',2X,4A1)
      WRITE(*,130)
      WRITE(*,510)
510   FORMAT('0',5X,'TO ASSIST IN ASSIGNING THE REST OF THE CHARACTERS',
     +/,5X,'ALL ITEMS WITH THE ABOVE FOUR(4) CHARACTERS CAN BE',/,5X,
     +'RETRIEVED FROM THE DATABASE')
      PAUSE 'NOTE DOWN THE ABOVE FOUR CODES! YOU NEED TO INPUT THEM'
      WRITE(*,520)
520   FORMAT('0',5X,'DO YOU WANT THE RETRIEVAL TO PROCEED? Y/N?')
      READ(*,'(A1)',ERR=530) ANS
        IF(ANS.EQ.ENT) CALL SORTIT
        CONTINUE
530   WRITE(*,540)
540   FORMAT('0',5X,'MORE ITEMS TO BE NUMBERED/ CODED? Y or N ?')
      READ(*,'(A1)',ERR=550) ANS
        IF(ANS.EQ.ENT) GOTO 5
        GOTO 560
550   CONTINUE
      PAUSE '******ENTRY ERROR ..   NO INPUT'
      GOTO 530
560   CONTINUE
      END
```

```
      SUBROUTINE DATPRO
C  ***************************************************************
C  **  THIS IS THE DATA PROCESSING SUBROUTINE, FOR              **
C  *   EVALUATING ECONOMIC ORDER QUANTITIES AND                 **
C  *   BUFFER STOCKS USING SIMPLE AS WELL AS  THE               **
C  **  SERVICE LEVEL, PROBABILISTIC APPROACH  FOR               **
C  **       NON-DETERMINISTIC DEMAND CASES.                     **
C  ***************************************************************
C
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DAT2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR

      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR
      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,AVDEM,
     &        EXPTAV,ORDRNO,ORDRDT,ORDRQT


      ICALNO=7
      CALL INTROD
10    CONTINUE
20    WRITE(*,30)
30      FORMAT('0','ENTER THE PART NUMBER FOR THE ITEM')
      READ(*,35,ERR=10) PARTNO
35    FORMAT(A)

      CALL DATVER
      WRITE(*,40)
40    FORMAT('0','WHICH EVALUATION APPROACH DO YOU INTEND TO USE ?',
     &/,'0','1....SIMPLE / BASIC EOQ FORMULA FOR ORDER SIZE.',/,' ',
     &'2....ROLEV CALCN.with SERVICE LEVEL- PROBABILISTIC APPROACH.',
     &/,'0','PLEASE ENTER ONE OF 1 or 2')

      READ(*,50) IOPTN
50    FORMAT(I1)
      IF(IOPTN.EQ.1) THEN
55       CONTINUE
      WRITE(*,60) COSTOD
60    FORMAT('0','THE RECORDED COST OF PLACING AN ORDER IS $',
     &F6.2,/,' ','WHAT ORDER COST WOULD YOU LIKE TO USE FOR THE',
     &' EVALUATION ?',/,' ','format F6.2')
      WRITE(*,70)
70    FORMAT('0','ENTER ALL BLANKS TO USE SAME FIGURE !')
```

```
      READ(*,80,ERR=55) AA
80    FORMAT(BN,F6.2)
      IF(AA.NE.0.) COSTOD=AA
90    CONTINUE
      WRITE(*,100) HOLCOS
100   FORMAT('0','THE RECORDED ANNUAL HOLDING COST PER ITEM IS $',
     &F6.2,/,' ','WHAT COST WOULD YOU LIKE TO USE ?  format F6.2')
      WRITE(*,70)
      READ(*,80,ERR=90) AA
      IF(AA.NE.0.) HOLCOS=AA
110   CONTINUE
      WRITE(*,120) AVDEM, UNIT
120   FORMAT('0','THE RECORDED AVER. ANNUAL DEMAND FOR THIS ITEM ',
     &'IS S',I8,A,/,' ','WHAT ANNUAL DEMAND SHOULD BE USED ?',
     &' format Integer 8')
      WRITE(*,70)
      READ(*,130,ERR=110) IA
130   FORMAT(BN,I8)
      IF(IA.NE.0) AVDEM=IA
140   CONTINUE
      WRITE(*,150) COST1
150   FORMAT('0','THE RECORDED UNIT COST IS $',F8.2,/,' ',
     &'WHAT COST IS TO BE USED ? format F8.2 ')
      WRITE(*,70)
      READ(*,160,ERR=140) AA
160   FORMAT(BN,F8.2)
      IF(AA.NE.0.) COST1=AA

      AA=SQRT(2.*COSTOD*REAL(AVDEM)/HOLCOS)
      IA=INT( SQRT(2.*COSTOD*HOLCOS*REAL(AVDEM)))

      WRITE(*,170) AA, UNIT, IA
170   FORMAT('0','THE CALCULATED ECONOMIC ORDER QUANTITY IS',F7.1,
     &A,/,'0','THE CALCULATED TOTAL INVENTORY [VARIABLE] COST IS $',
     &I7,4X,'[to the nearest dollar]')

      ELSEIF( IOPTN.EQ.2 ) THEN
            WRITE(*,180)
180      FORMAT('0','THE EVALUATION OF BUFFER/SAFETY STOCK IS ',
     &'BASED ON AN ASSUMPTION',/,' ','OF CONSTANT LEAD TIME AND',
     &' PROBABILISTIC DEMAND OVER THE LEAD TIME.',/,' ','IF LEAD ',
     &'TIME CAN NOT BE CONSIDERED CONSTANT ONE SHOULD PLAY',/,' ',
     &'SAFE BY SOME INCREASE IN THE SERVICE LEVEL FIGURE !')
      PAUSE 'NORMAL DEMAND DISTR. OVER LEAD TIME IS EMPLOYED.'
190   CONTINUE
      WRITE(*,200) PLDTIM
200   FORMAT('0','THE RECORDED PROCUREMENT LEAD TIME IS ',I4,
     &' days.',/,' ','ENTER THE APPROX. LEAD TIME IN *WEEKS* TO',
     &' BE USED FOR THE',/,' ','EVALUATION. format Int.2')
      READ(*,210,ERR=190) PLDTIM
210   FORMAT(BN,I2)
      AA=REAL(PLDTIM)*REAL(AVDEM)/52.
215   CONTINUE
      WRITE(*,220) PLDTIM, AA, AVDEM
```

```
220   FORMAT('0','THE AVERAGE ',I3,' WEEK [lead time] DEMAND IS',
     &F7.1,/,' ','FROM THE AVERAGE ANNUAL DEMAND OF',I8)
      WRITE(*,230)
230   FORMAT('0','ENTER LEAD TIME DEMAND FIGURE TO BE USED -',
     &'format I 5')
      WRITE(*,70)
      READ(*,240,ERR=215) IA
      IF(IA.EQ.0) IA=INT(AA)
240   FORMAT(BN,I5)
245   CONTINUE
      WRITE(*,250) SEVLEV
250   FORMAT('0','THE RECORDED [customer] SERVICE LEVEL IS',F4.2,
     &/,' ','ENTER SERVICE LEVEL FOR CALCULATION OF SAFETY STOCK',
     &' format F4.2')
      WRITE(*,70)
      READ(*,260,ERR=245) AA
260   FORMAT(BN,F4.2)
      IF( AA.NE.0.) SEVLEV=AA
270   CONTINUE
      WRITE(*,280)
280   FORMAT('0','A LEAD TIME DEMAND STANDARD DEVIATION FIGURE IS',
     &' REQUIRED FOR THIS',/,' ','EVALUATION. YOU MAY WANT TO EXIT ',
     &'FOR A WHILE IF YOU DO NOT HAVE THIS FIGURE !')
      WRITE(*,290) PLDTIM, IA
290   FORMAT('0','DO YOU HAVE THE STANDARD DEVIATION FIGURE FOR THE',
     &I3,' WEEK',/,' ','PERIOD - ',I5,' UNIT AVERAGE DEMAND ? Y or N?')
      READ(*,300) ANS
300   FORMAT(A)
      IF(ANS.EQ.NO) GOTO 350
310   CONTINUE
      WRITE(*,320) IA,PLDTIM
320   FORMAT('0','ENTER THE STAND. DEVIATION FIGURE FOR THE ',I5,
     &' UNIT AVERAGE',/,' ','DEMAND OVER THE ',I3,' WEEK LEAD TIME.',
     &' format F8.2')

      READ(*,160,ERR=310) AA

      IF(SEVLEV.GT.0.999) THEN
            AFACTR=3.09
      ELSEIF(SEVLEV.GT.0.995) THEN
            AFACTR=2.576
      ELSEIF(SEVLEV.GT.0.99) THEN
            AFACTR=2.326
      ELSEIF(SEVLEV.GT.0.975) THEN
            AFACTR=1.960
      ELSEIF(SEVLEV.GT.0.95) THEN
            AFACTR=1.645
      ELSEIF(SEVLEV.GT.0.900) THEN
            AFACTR=1.282
      ELSEIF(SEVLEV.GT.0.85) THEN
            AFACTR=1.036
      ELSEIF(SEVLEV.GT.0.80) THEN
            AFACTR=0.842
      ELSEIF(SEVLEV.GT.0.75) THEN
```

```
            AFACTR=0.674
      ELSEIF(SEVLEV.GT.0.70) THEN
            AFACTR=0.524
      ELSEIF(SEVLEV.GT.0.60) THEN
            AFACTR=0.300
      ELSEIF(SEVLEV.GT.0.50) THEN
            AFACTR=0.125
            PAUSE 'VERY LOW SERVICE LEVEL -- 0.55 USED !'
      ENDIF

      SSTOCK=AFACTR*AA
      ROLEV=IA+INT(SSTOCK)+1


      WRITE(*,330) SSTOCK,ROLEV
330   FORMAT('0','THE CALCULATED BUFFER/SAFETY STOCK IS',F7.1,/,' ',
     &'THEREFORE FROM THE AVERAGE LEAD TIME DEMAND THE RE-ORDER LEVEL',
     &/,' ','SHOULD BE:',I5,' UNITS.')
350   CONTINUE
       ENDIF

      WRITE(*,360)
360   FORMAT('0','PLEASE NOTE THAT:',/,18('*'),/,'0','THE USEFULNESS',
     &'SS AND/OR RELIABILITY OF THIS EVALUATION ',/,' ','VERY MUCH',
     &' DEPENDS ON THE EXTENT TO WHICH THE VALUES OF THE',/,' ',
     &'VARIABLES USED/ENTERED , APPROXIMATE THE REAL VALUES !')
      WRITE(*,370)
370   FORMAT('0','ARE YOU DOING ANOTHER EVALUATION ? Y or N ?')
      READ(*,300) ANS
      IF(ANS.EQ.YES) GOTO 10
      CONTINUE
      END
```



```
      PROGRAM MAIN27
C  ***********************************************************************
C  **
C  **  THIS MAIN PROGRAM CONTROLS THE EXECUTION OF TWO IMPORTANT     **
C  **  SYSTEM ROUTINES - THE DATA PROCESSING SUB-SYSTEM AND THE      **
C  **  REGULAR REPORT GENERATING SUBROUTINE..        **
C  **
C  ***********************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
      CHARACTER*1 ANS,NO,YES
      CHARACTER*2 USERCD
      CHARACTER*8 PARTNO,DATE,INIT
      CHARACTER*36 PAR

      NO='N'
      YES='Y'

      DO 10 I=1,10
         MM(I)=0
10    CONTINUE

      CALL ENTRES
         IF( MM(4).EQ.3) GOTO 100

      CALL EXCEPT
20    CONTINUE
        WRITE(*,30)
30    FORMAT('0','THIS PROGRAM HANDLES THE FOLLOWING ROUTINES=>',/,
     &'0','1....DATA PROCESSING / EVALUATION SUB-SYSTEM:',/,' ',
     &'2....REGULAR REPORT GENERATION:',/,'0','ENTER ANY OTHER DIGIT',
     &' TO EXIT.',/,'0','===>ENTER ONE OF THE ABOVE.!')
      READ(*,40,ERR=20) MM(1)
40    FORMAT(I1)
      IF( MM(1).EQ.1) THEN
            CALL DATPRO
        ELSEIF( MM(1).EQ.2) THEN
            CALL REGREP
        ELSE
            GOTO 100
      ENDIF

      GO TO 20
100   CONTINUE
      END
```

```
      SUBROUTINE REGREP
C     ***************************************************************************
C     **
C     **  THIS IS THE REGULAR REPORT GENERATION SUB-ROUTINE...      **
C     **
C     ***************************************************************************
      COMMON /C1/ ICALNO,MM(10),MODE,IRECNO
      COMMON /C2/ HOLCOS,COSTOD,COST1,SEVLEV,EOQTY,ROLEV,PLDTIM,
     &            AVDEM,INVLEV,SOHAND
      COMMON /C3/ ORDRNO(3),ORDRDT(3),ORDRQT(3),EXPTAV(3)
      COMMON /C4/ NAME,UNIT,MOB,CLASS,DEPCOD,CODLOC,LOCNAM
      COMMON /C5/ DATE1,DATE2,DATEA,DATEB
      COMMON /C6/ ANS,NO,YES,PARTNO,DATE,INIT,USERCD,PAR
      CHARACTER*1 ANS,NO,YES,MOB,CLASS
      CHARACTER*2 CODLOC,USERCD
      CHARACTER*4 UNIT
      CHARACTER*5 DEPCOD
      CHARACTER*8 PARTNO,DATE,INIT,BLK,LOCNAM
      CHARACTER*20 NAME
      CHARACTER*36 PAR
      INTEGER DATE1,DATE2,DATEA,DATEB,EOQTY,ROLEV,PLDTIM,EXPTAV,
     &        AVDEM,ORDRNO,ORDRDT,ORDRQT

      ICALNO=9
      CALL INTROD

      PAUSE 'PLEASE CONTROL PRTSC [Ctrl Prtsc] FOR PRINTER REPORT'
      WRITE(*,5)
5     FORMAT('0',' ',/,'0',' ')
      CONTINUE
      PAR='** R E G U  L A R    R E P O R T ** '
         CALL THEAD
         CALL RTHEAD
      IRECNO=0

      OPEN(4,FILE='DATA1.INV',ACCESS='DIRECT',RECL=72)
      OPEN(5,FILE='DATA2.INV',ACCESS='DIRECT',RECL=72)
      OPEN(6,FILE='DATA3.INV',ACCESS='DIRECT',RECL=72)

10       IRECNO=IRECNO+1
      IF( MM(7).EQ.0) THEN

       READ(4,20,REC=IRECNO,END=50) PARTNO,NAME,UNIT,MOB,CLASS,
     &DEPCOD,DATE1,DATE2,CODLOC,LOCNAM
       READ(5,30,REC=IRECNO,END=50) SOHAND,HOLCOS,COSTOD,COST1,
     &SEVLEV,EOQTY,ROLEV,PLDTIM,AVDEM,INVLEV
       READ(6,40,REC=IRECNO,END=50) ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &ORDRQT(1),ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),
     &ORDRDT(3),EXPTAV(3),ORDRQT(3)

20     FORMAT(A8,1X,A20,1X,A4,2(1X,A1),1X,A5,2(1X,I6),1X,A2,1X,A8)
30     FORMAT(F8.2,2(1X,F6.2),1X,F8.2,1X,F4.2,2(1X,I5),1X,I4,1X,I8,
```

```
     &1X,13)
 40   FORMAT(15,2(1X,14),2(1X,15),2(1X,14),2(1X,15),2(1X,14),1X,15)
      GOTO 55
 50   CONTINUE
      MM(7)=IRLCNO-1
 55   CONTINUE

      WRITE(*,60) PARTNO,NAME
 60   FORMAT('0','PART NUMBER:',3X,A,/,13X,11('='),/,'0',
     &'PART DESCRIPTION:',2X,A)
      WRITE(*,70) SOHAND,COST1,HOLCOS,COSTOD,EOQTY,ROLEV,PLDTIM
 70   FORMAT('0',' ON  HAND ',' UNIT COST',' HOL COST ',' ORD COST ',
     &'EC.ORD.QTY',' RE ORD LE ','PROC/LTIME',/,2X,8('*'),2X,9('*'),
     &1X,8('*'),2X,8('*'),1X,10('*'),1X,9('*'),1X,10('*'),/,' ',
     &4(F10.2),3(I10))

      WRITE(*,100) AVDEM,SEVLEV,INVLEV,MOB,CLASS,UNIT,DEPCOD
 100  FORMAT(' ',' AVER DEM',' SERV LEV',' INV LEV',' M OR BUY',
     &' INV CLASS',' MEAS UNIT',' DEPT COD',/,' ',2X,8('*'),2X,8('*'),
     &3X,7('*'),2X,8('*'),1X,9('*'),1X,9('*'),2X,8('*'),9X,/,' ',I10,
     &F10.2,I10,9X,A,9X,A,6X,A,5X,A)

      WRITE(*,120) CODLOC,LOCNAM,ORDRNO(1),ORDRDT(1),EXPTAV(1),
     &ORDRQT(1)
 120  FORMAT(' ',' LOCN CODE',' LOCN NAME',' ORD NO 1:',' ORD 1 DAT',
     &' EXP. ARR ',' ORD.1 QTY',/,' ',1X,9('*'),1X,9('*'),1X,9('*'),
     &1X,9('*'),1X,8('*'),2X,9('*'),/,' ',7X,A,3X,A,2X,I7,3(I10))
      WRITE(*,140)ORDRNO(2),ORDRDT(2),EXPTAV(2),ORDRQT(2),ORDRNO(3),
     &ORDRDT(3),EXPTAV(3),ORDRQT(3)
 140  FORMAT(' ',' ORD # 2','   DATE:',' EXP.ARR','  QUATTY',
     &' ORD # 3','   DATE:',' EXP.ARR','  QUATTY',/,' ',1X,7('*'),3X,
     &5('*'),1X,7('*'),2X,6('*'),1X,7('*'),3X,5('*'),1X,7('*'),2X,
     &6('*'),/,' ',8(I8))

       ELSE
      MM(7)=0
      GOTO 150
       ENDIF

      GOTO 10
 150    CONTINUE
      WRITE(*,160)
 160  FORMAT('0',25('*'),' END ',25('*'))
      END
```

APPENDIX D - Running the Computer Programs

All the programs for the inventory control system are fully interactive, thus once accessed they continually prompt the user for the necessary information, program options etc., as well as provide the applicable format for the information to be input.

Getting the IBM micro-computer system started.

Before one can run an execution program, the computer system must be initiated so that it is ready for this purpose. An illustration of the step by step procedure in starting the system follows below.

Bold print represents text and prompts from the computer system, while *italic print* signifies input from the user. "Disk drive system" refers to the micro-computer main unit, while drive "A" and drive "B" are the left side drive and the right side drive respectively. Comments are placed in brackets.

1. Insert the disk operating system (DOS) diskette in disk drive A. (this is the drive to the operator's left)
2. Switch on the system main switch - on the side of the disk drive system. (right side of the operator!)
   (Wait for the computer to start. It takes some 45 seconds or so before a prompt appears.)
   Enter today's date (m-d-y):
   (the date may be entered in the form '6/15/83', '6/15/1983', '6-15-83', or '6-15-1983', noting that the

entries are in the order - month, day and year. If an unacceptable entry is input, the prompt will be displayed again for another entry. Press the RETURN button[◄┘] after the date entry.)

```
Enter today's date (m-d-y):6/15/83
The IBM Personal Computer DOS
Version 1.00 (C) Copyright IBM Corp 1981
A>_
```

(The system is now ready.)

3. Remove the disk operating system (DOS) diskette from disk drive A.

To run any one of the execution programs, proceed according to the procedure given in the following section. Each execution (run) file resides on a separate diskette, which is labelled accordingly. Depending on the system application required, the appropriate execution program should be chosen from the three system programs described in the preceding sections.

To access the execution files.

The three execution files (for programs MAIN11, MAIN22 and MAIN33) are respectively named MSC11.EXE, MSC22.EXE, & MSC33.EXE. In order to run any particular program, the corresponding program diskette (for MSC11.EXE, MSC22.EXE, or MSC33.EXE) and the data diskette are required. During execution, the data diskette should be placed in the default disc drive - the drive, A or B appearing on the left

side of the screen, A> or B>.

In the following explanation it is assumed that the computer is running from disc drive A.

NB: Whenever an entry (command) is made to the system the return key should be pressed at the end of the entry to indicate end of command.

Procedure:

1. Place data diskette in drive A
2. Place program diskette in drive B
3. Enter B:MSC11 (or B:MSC22, or B:MSC33 depending on the execution program - run program being accessed).

The run file is loaded into computer memory and execution begins. Thereafter, one only needs to input information as prompted by the computer, noting carefully the data formats that are required, which are also provided by the system.

Important note on data input.

To make the process of interactive data input into the system more convenient, all the formats in the programs incorporate the option for ignoring trailing blanks - ie. formats are in the form "format(BN,fM.N)" or "format(BN,I8)" etc. The effect of incorporating this option is that all blanks following any digits in I or F format are ignored. They are not treated as zeros as is the case with normal formats. This allows the user to input only the necessary digits for any format and only fill the remaining field with

blanks. Thus, if for instance the format is I6, to input the number "46" one only needs to punch in a 4 and a 6 followed by four blanks or more. The need for a right justified input, which is very inconvenient, is therefore eliminated. One notes that if this option was not in effect the preceding entry would be intepreted as "460000"!

The rule of thumb is to input the necessary characters, including all necessary zeros, and to fill the rest of the field with 'at least the minimum number of blanks required to meet format requirements'. The system allows infinite blank padding at the end of formatted records so there is no harm in inputting more blan

Example:

To enter 456 when the format is I8 (8 digits).

Enter 4,5, and 6 followed by five or more blanks.

The same applies to F formats even when the decimal point is not entered.

The following pages show the initial step by step procedures in running the three system programs.

Start-up Procedures for Program Execution

Program # 1: [MSC11.EXE]

1. :Start the Computer System following the procedure in section 4.5.

```
Enter today's date (A-d-y): 6/27/83

The IBM Personal Computer DOS
Version 1.00 (C)Copyright IBM Corp 1981
```

2. Insert data diskette in drive A, and program diskette [MSC11.EXE] in drive B.

3. Proceed as follows:

```
A>B:MSC11

          ************************************************
          **                                            **
          **  ARGO ENGINEERING LIMITED      DATE:(M/D/Y)  **
          **                                  XXXXXXXX  **
          **  THIS IS THE:                              **
          **  --->>    MAIN   TRANSACTION   PROGRAM     **
          **                                            **
          ************************************************

 THE DEPARTMENT OF MECHANICAL ENGINEERING-- UNIVERSITY OF ALTA.
 ------------------------------------------------------------


 PLEASE ENTER YOUR INITIALS (8 CHARACTERS)



 EM.KOMBE

 ENTER TODAY'S DATE (M/D/Y -eg.04/22/83)
 6/  06/27/83

 ENTER YOUR PERSONAL USER ID. CODE
 AB

 RECORDED DATES IN DATABASE== 62783 &  627 (m/d/yr)

 DO YOU WANT EXCEPTION REPORT PRINTED ? Y/N?
 N
```

(exception report skipped!)

```
**************************************************
**                                              **
**  ARGO ENGINEERING LIMITED       DATE:(M/D/Y) **
**                                    06/27/83  **
**  THIS IS THE:                                **
**  --->>   TRANSACTION   OPTIONS   LISTING     **
**                                              **
**************************************************

THE FOLLOWING SUBROUTINES ARE CURRENTLY AVAILABLE
PLEASE ENTER ONE OF THE RELEVANT OPTIONS DEPENDING ON THE INTENDED
USE OF THE SYSTEM.

OR ENTER NINE (9) TO EXIT!

 1....GENERAL TRANSACTIONS, INPUT/OUTPUT ETC.
 2....ENQUIRY---> ANY DATA OR PARTICULAR INFORMATION.
 3....ORDERS,--PROCESSING OF ORDERS, PLACEMENT/RECEIPT
 5....SORTING FOR PART ID OF GIVEN SIGNIF. CODES:

ENTER THE APPROPRIATE OPTION:
```

4. Choose one of the above options for execution - and proceed.

```
**************************************************
**                                              **
**  ARGO ENGINEERING LIMITED       DATE:(M/D/Y) **
**                                    06/27/83  **
**  THIS IS THE:                                **
**  --->> INPUT/OUTPUT AND ALTERATIONS ROUTINE  **
**                                              **
**************************************************

IN THE FOLLOWING PROMPT ENTER BLANKS FOR THE PART
NUMBER IF YOU WANT TO EXIT THE TRANSACTIONS SUBROUTINE.
NEW PART NUMBERS MAY BE INPUT !
Press <enter> to continue.

OPLEASE (RE):ENTER  THE  INVENTORY IDENTIFICn #(PARTNO)-[AB]
A258    A14B2969


INVENTORY RECORD NUMBER=   2
----------------------------


  ENTER TWO BLANKS (OO) TO EXIT !

 11...WITHDRAWAL OF INVENTORY or returns(-ve)
 12...INVENTORY RECEIPTS - ORDER ARRIVALS ONLY !
 13...HOLDING COSTS (HOLCOS)
 14...ORDERING COSTS (COSTOD)
 15...INVENT. ITEM COST (COST1)
 16...SERVICE LEVEL (SERLEV)
 17...ECONOMIC ORDER QUANTITY(EOQTY)
 18...RE-ORDER LEVEL(ROLEV)
 19...PROCUREMENT LEAD TIME (PLDTIM)
 22...AVERAGE ANNUAL DEMAND (AVDEM)
 23...INVENTORY LEVEL(S), (INVLEV)
```

```
***** PICK THE RELEVANT CODE *****
Press <enter> to continue.

 24...CHANGE OF PART NUMBER (PARTNO)
25...PART DESCRIPTION--NAME
26...INVENTORY MEASURE UNIT
27...MAKE OR BUY  (MOB)
28...INVENTORY CLASS (A/B/C/)
29...USER DEPARTMENT CODE (DEPCOD)
35...LOCATION CODE (CODLOC)
36...LOCATION NAME  (LOCNAM)

ENTER THE CODE# FOR THE DESIRED TRANSACTION (12)
ON THE ITEM NUMBER A14B2969
11
```

Program # 2: [MSC??.EXE]

1. Start the system following the procedure as in section 4.5.
2. Insert data diskette in drive A and program diskette [MSC22.EXE] in drive B.
3. Proceed as follows:

```
A>B:MSC22

THE DEPARTMENT OF MECHANICAL ENGINEERING-- UNIVERSITY OF
==========================================================


PLEASE ENTER YOUR INITIALS (8 CHARACTERS)
EM.KOMBE

ENTER TODAY'S DATE (M/D/Y -eg.04/22/83)
06/27/83

ENTER YOUR PERSONAL USER ID. CODE




ENTER TODAY'S DATE (M/D/Y -eg.04/22/83)
06/27/83

ENTER YOUR PERSONAL USER ID. CODE
A9

RECORDED DATES IN DATABASE-- 62783 &  627 (m/d/yr)

DO YOU WANT EXCEPTION REPORT PRINTED ? Y/N?
N
```

```
THIS PROGRAM HANDLES THE FOLLOWING ROUTINES->

1....DATA PROCESSING / EVALUATION SUB-SYSTEM:
2....REGULAR REPORT GENERATION:

ENTER ANY OTHER DIGIT TO EXIT.

--->ENTER ONE OF THE ABOVE.!
4

A>
```

4. Choose one of the above options for execution -- and proceed.

Program # 3: [MSC33.EXE]

1. Same procedure as above, except using program MSC33.EXE instead.
2. After inserting diskettes, proceed as follows:

```
B:MSC33

THE DEPARTMENT OF MECHANICAL ENGINEERING-- UNIVERSITY OF ALTA.
-------------------------------------------------------------


PLEASE ENTER YOUR INITIALS (8 CHARACTERS)
EM.KOMBE

ENTER TODAY'S DATE (M/D/Y -eg.04/22/83)
06/27/83

ENTER YOUR PERSONAL USER ID. CODE
A7

RECORDED DATES IN DATABASE-- 42783 &  627 (m/d/yr)

IF SORTING IS REQUIRED-->ENTER 2

FOR CODING ENTER ANY OTHER DIGIT !
```

3. Continue reacting to system prompts as required.

```
3

**************************************************************
**
**  THIS IS THE INVENTORY IDENTIFICATION PROGRAM. IT IS AN AID
**  IN THE ASSIGNMENT OF PART IDS. AS AN OPTION
**  SORTING (SUBROUTINE SORT) CAN BE DONE FOR OTHER ITEMS WITH
**  SIMILAR CHARACTERS TO AVOID ANY ID. DUPLICATION..
**
**************************************************************

FOLLOW THE FOLLOWING PROMPTS CAREFULLY TO DETERMINE WHICH
SIGNIFICANT CODES SHOULD BE ASSIGNED TO THE ITEM IN QUESTION:
THE SERIES OF PROMPTS FOLLOW FROM HERE !!
Press <enter> to continue.

          FINAL PRODUCT /SPECIAL INVENTORY CLASS CODE
         ---------------------------------------------
          WOULD YOU CONSIDER THE ITEM IN QUESTION TO
         BELONG TO ONE OF THE FOLLOWING SPECIAL CLASSES?
         IF SO ENTER THE APPROPRIATE LETTER - ELSE THEN
         ENTER A BLANK
              E: ELECTRICAL
              H: HYDRAULICS AND PNEUMATICS
              T: TOOLS
              X: STANDARD ITEMS/COMMERCIALS (eg.BOLTS & NUTS)
              ENTER A BLANK FOR NONE OF THE ABOVE:



WOULD YOU CONSIDER THE ITEM TO BE PARTICULARLY
USED FOR ONE OF THE COMPANY'S PRODUCTS AS OUTLINED BELOW?
THIS APPLIES ALSO FOR AN ITEM THAT IS MAINLY USED
FOR A PARTICULAR PRODUCT BUT ALSO USED ..TO A LESS
EXTENT IN ANOTHER/OTHERS.
                THE PRODUCTS LIST NOW FOLLOWS
Pause.
Press <enter> to continue.
```

NB: In all the programs the system will jump back to the beginning of program after a subroutine's execution is completed, so that one can pick another routine if desired. Unless there is a "user" error, the program will only terminate when the user indicates so.