1-1-1977

A general purpose material requirements planning system.

Mark Pryor

A GENERAL PURPOSE
MATERIAL REQUIREMENTS
PLANNING SYSTEM

by

Mark Pryor

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1977

CERTIFICATE OF APPROVAL

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

15 December 1977
Date

Professor in Charge

Chairman of the Department of Industrial Engineering

ACKNOWLEDGEMENTS

I wish to thank Ingersoll-Rand for the manufacturing background, and Professor Emory W. Zimmers, Jr. for the topic of this thesis.

I wish to thank my parents for their assistance and support throughout my education.

I especially wish to express my gratitude and admiration to my wife, Diane, who stood beside me and encouraged me to continue my academic pursuits.

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

The objective of this thesis is to develop a general purpose material requirements planning (MRP) system. The system developed is intended to allow experimentation and education in this area.

As part of this thesis, the traditional operations research method of economic order quantity is compared to material requirements planning. This establishes the purpose for developing such a system.

The package contains a data base management system, an inventory accounting system, an engineering information system, an MRP system, and several utilities. A demonstration of the investigative capabilities of the package is shown.

The appendix consists of a self-contained manual which, in conjunction with the software, can be used to become familiar with MRP, and to learn and use the system on the Lehigh University computer.

I. INTRODUCTION

Manufacturing companies require fairly rigid control over inventory, since it accounts for a large portion of their capital investment. Consequently, work orientation in industry is significantly different from that in an academic environment.

Material Requirement Planning (MRP) is an inventory management system used in the more progressive manufacturing companies, and little work is found concerning this method in educational institutions. Although many MRP systems exist today, they are usually highly customized, understood by only a few people, and are not easy to use for experimentation. The MRP system developed for this thesis can be used to illustrate basic concepts and material problems. It can also be used to investigate new ideas, simulate potential projects, and even to do high level production planning.

Enclosed is a complete, self-contained, manual for learning MRP and for using the package on the Lehigh University computer. This basic package has much potential for future expansion.

II. INVENTORY PROBLEMS

Even the simplest inventory systems have problems which must be overcome. A one-product, retail operation must be concerned with demand for the product and lead time to acquire it. If the demand is positively known, it is simple to project when stock will be depleted; if lead time is positively known, it is simple to decide when to reorder stock to cover the stock-out. If one of these items is not positively known, problems arise. If demand is known, but lead time is not, the stock-out may occur when planned, but the stock may not arrive on time to cover it; if lead time is known, but demand is not, the stock may arrive on schedule, but a premature stock-out may occur due to unusual demand. The simplest solution to this problem is to carry extra product, known as safety stock.

A multiple product, retail operation must handle the same problem for each product; but all items contend for a finite inventory space, and so safety stocks can not be too high.

When assembly is required, each complete product has the same problems as a retail operation. Additional complications arise depending upon the type of demand. If all parts are made to order, given full lead time (time to purchase and assemble the complete product), things are not too difficult. If parts are made to stock, to supply demand in less than full lead time, a production plan must be established which anticipates true demand. Many companies must make items to order, to stock and to supply spare parts demand.

All of these problems relate to a simple conflict, that of service level vs. cost. Specifically, it is important to supply stock for all demand (to have a high storage level), but to minimize cost (inventory investment, storage costs, etc.).

III. ECONOMIC ORDER QUANTITY - THE CLASSIC APPROACH

The classic approach to these problems is the Economic Order Quantity method. This makes several assumptions concerning demand and lead time. As typified by Plane[4] and Johnson[5], demand is assumed fixed, or continuous, and independent of any other demands. Lead times are assumed reasonably fixed and fairly accurate. These assumptions are used to develop the model.

The method of implementing EOQ is to reorder some economic order quantity whenever the stock level drops to a reorder point. Figure 18 demonstrates the theoretical inventory model. Assume that initial inventory level is 50, the lead time is one month, and the demand is constant as shown in the top graph. It follows that the demand over lead time is five. Assume that a safety stock level has been determined to be ten, then the reorder point will be 15 (demand over lead time + safety stock). Further, assume that the costs were such that an EOQ of 60 was chosen.

Following the bottom chart, inventory level begins at 50, becomes steadily depleted by a constant demand, until seven months pass. At this time an order is initiated, and one month later it arrives. This is shown by the inventory level continuing to drop until month eight, when 60 parts (the EOQ) arrive and the cycle begins again. Note that the stock never dropped below ten (the safety stock).

Any fluctuation in demand or lead time should be handled by the safety stock allowed. The only problems are to determine safety stock for high service level and to determine the EOQ which minimizes cost.

This requires some data in addition to the inventory level. The demand must be determined by some method, such as exponential smoothing. The lead time must be assigned based upon the service level desired (subjective) and demand or lead time variation (using mean absolute deviation, or some similar method).

Finally, several costs must be determined in order to express total cost. The carrying cost is the cost of carrying the inventory. The set-up cost is the cost for a single order, such as machine set-up and paperwork costs. The stock-out cost is a very subjective formula which expresses opportunity costs, cost of customer bad will, cost of backordering, etc.

IV. ECONOMIC ORDER QUANTITY PROBLEMS

Several problems exist when using the traditional model. Joseph Orlicky[1] is credited with the concept of dependent demand. Stated simply, not all demand for parts is independent of other demand.

It is quite obvious that the demand for tricycles has an effect on the demand for tricycle tires. Only the spare parts demand for the tires is reasonably independent of the demand for tricycles. In order to manufacture the tricycles to meet the demand, several tires will be needed for production.

The concept of a reasonably fixed lead time, whose estimate is important, falls apart under closer examination, as discussed by Wight[2] and Harrington[3]. If a part normally requires 15 weeks to make, it probably could be done in five weeks if necessary (and one week if the plant manager wanted it). Of course other jobs would suffer if we continually cut the lead time. Lead time is a function of priority, and should be used as such; that is, important jobs should be done more quickly, and unimportant jobs need not be done within normal lead time.

Costs are a weak part of the data used by the model. Although set-up costs may be determined, stock-out costs are totally subjective and highly suspicious. It may seem that carrying costs may be determined but, as Wight[2] points out, inventory costs are incurred in the aggregate; and, therefore, minimum costs could only be developed if all similar parts were lot-sized as a group.

EOQ and similar methods are basically order launching systems. They devote their efforts to setting up orders and leaving it alone after that. Actually, priorities change after an order is initiated, and it may need rescheduling or even cancellation.

These systems may generate an unreasonable quantity, due to lack of practical limitations such as size. Although the main selling point of EOQ is the "perfect" quantity, reasonable quantities may be developed much cheaper without the system, using an intelligent guess. Wight[2] says that these systems often generate "the right quantity at the wrong time".

Demand is normally lumpy; that is, it has seasonal fluctuations. There are large customer orders such as contracts, and demand is lot-sized by the customer. (They have inventory problems, too).

Safety stock, an integral part of EOQ, is expensive. It represents uncertainty, but many variances are not due to uncertainty; they are due to well known reasons, such as major contracts.

Figure 19 demonstrates what may actually occur when following an EOQ model. As in Figure 18, assume an initial inventory level of 50, a lead time of one month, an <u>average</u> demand of five per month, safety stock of ten, reorder point of 15, and EOQ of 60. Although the average demand is five per month, this may occur as regular demands of 30 every six months, as shown in the top chart.

Following the bottom chart, inventory remains at the initial 50 until month three, when 30 are used. For the next six months, inventory remains at 20 (above the reorder point). At month nine, a demand of 30 wipes out inventory and creates a stock-out of ten pieces. We must put in an emergency order for 60 pieces (the EOQ). When the stock arrives, we satisfy the backorder for ten (unless it was cancelled) and stock the remaining 50. This begins another identical cycle.

It is possible that this demand is well known, but the ordering policy causes excess inventory to be carried for months (from months zero to nine), an unnecessary stock-out to occur, and an emergency order for more stock than necessary. It is interesting to note that the probable solution to this problem is to raise the safety stock level.

Statistics may also be utilized to show the impracticality of EOQ for assembly plants. If ten parts are needed simultaneously to assemble an item, then consider the probability of assembly for various service levels of each part. If a service level of 98% is possible for each part, then the probability of all ten parts being available at the same time is $(.98)^{10} = 80\%$. If the service level is only 90% per part, we could only assemble the item $(.90)^{10} \cong 35\%$ of the time.

Perhaps the most major specific failing of the system is that users find out that often times they can do better with an informal system. At this point, the formal EOQ system becomes just a system which must be circumvented. The informal system takes over the formal one completely.

V. THE MATERIAL REQUIREMENTS PLANNING APPROACH

The ideal solution to these problems would be to design a network which describes the problem, and to optimize it. Johnson[5] shows a very simple example of this, and it is obvious that a few thousand related parts would make this an intractable problem.

Material Requirements Planning is a simulation of stock usage with perfect rescheduling. It uses a discrete demand, which may include dependent demand. Lead times are given as average, and they are assumed flexible, depending upon need. Safety stock is discouraged in most cases.

Order launching is part of the system, but only to initially set up an order based upon current conditions. Once orders are launched, they are continually re-evaluated to expedite, deexpedite, or even cancel as conditions change. Once an order is originally set up, the want date is less important than the need date.

The demand in Figure 19, on the ideal MRP system, would be handled by simply ordering 30 (the projected demand) one month (the lead time) before it is needed; that is, at months 2,8,14,etc.

VI. THE MATERIAL REQUIREMENTS PLANNING PACKAGE

The MRP package described in the manual (see appendix) is a good teaching and investigative tool. The manual is designed to teach both the use of the package and the principles of MRP. A sample data base lays a good foundation to use the package for investigations.

The design of the system was influenced by the good and bad points of various systems encountered by the author in industry [6,7,8,9]. It is intended to be easy to understand, easy to use, and expandable. Many important features are not included (for example, lot sizing and allocations), but they are not necessary for a basic system.

Although a major use may be tutorial, this is not its only use. Management frequently has problems following conceptual arguments. A demonstration of inventory problems and solutions could well validate new systems or subsystems. New problems could be investigated and new solutions simulated before or after system design. A miniature data base, with critical or major items, could be manipulated for management level simulations and "what if" questions, before policy or master schedules are determined.

VII. SYSTEM ANALYSIS

Many factors were considered before the actual design of the software. A single multi-function program makes the use of the system less complicated. Interactive programming was chosen because it allows adaptive processing by the user; errors detected may influence future transactions, and inquiries could be used before continuing.

The choice of languages was influenced by three objectives:

> Complete programmatic control of data format is desired.
> Segmentation of code is necessary due to the program size.
> The source code should be reasonably portable for implementation on other computer systems.

Portability reduced the choices to COBOL, FORTRAN, and BASIC. Data format control eliminated BASIC. While program control of data format is possible in FORTRAN, the methods to implement this are peculiar to each computer system, and this violates portability. COBOL was the only remaining language.

It was desired to allow the data base to be stored and retrieved in a manner which will allow normal backup and recovery procedures. A very common storage medium, cards, requires 80 column records, so this was adopted.

Simplicity of operation may be accomplished by making each input act as a stand-alone transaction. Specifically, each input is processed completely before any other may be allowed. Additionally, a free-form format is used to reduce potential typing errors. A logical definition of input is accomplished by dividing them into four basic types, and then subdividing each type.

Program size requires all efforts possible to reduce core requirements; this is necessary because the Lehigh University system and others are not virtual. Data must be compacted and work areas must serve multiple purposes. Executable code must be thoroughly segmented.

Program complexity requires modularity and simple mainline structure. Transaction oriented processing may be accomplished by always completing execution with a branch to one common entry point. This entry point should be the beginning of transaction processing. Commonly used subroutines should be coded once and available to any other routine.

A working MRP system does not require much normally useful data, such as cost. Since the package is expandable, a minimum working data base is more desireable than a complete one that takes longer to implement.

Inquiry should be available in three basic forms:

Specific items searched and shown.
Complete file shown.
A useful subset of the complete file shown.

Most reports are simple in concept, but the stock status report is a notable exception. The MRP matrix must be clear and complete in describing the netting and offsetting process.

VIII. SYSTEM DESIGN

The MRP system developed is contained entirely within a single COBOL program. Figure 20 illustrates the file relationships. MRPIN is the disk file which may contain a data base to be used with the LOAD command. MPROUT is the disk file which will contain a data base after using the STORE command. All other input comes from the file named INPUT, and all reports and messages are directed to the file named OUTPUT.

The data bases are binary files of a very specific format. All records are 80 characters in order to make them compatable with cards. The first record contains all control information including record counts. The rest of the records are either item master, order, or chain records, with just enough records to contain the data present. Three records are required for each item master, but two orders fit into one record, and four chains fit into one.

In order to compress space, quantities have been limited to 4095 so that they require only two bytes each. Dates have been compressed into three bytes. The bits of the date contain the following data:

bits 1-9: corrected year
bits 10-13: month
bits 14-18: day

The year in a date of the first century in the calendar is unchanged, but a date in the following century has 100 added to its year. This allows dates to be directly compared to one another.

While a data base is being used, it is contained entirely in core. The control information is stored similar to any normal working data. Item masters, orders, and chains each occupy an array. The date file occupies another array, but it is not stored or loaded with the other data. Various pointers and counters are also used to manipulate the data.

A utility area exists which is used for sorts, reorganizations (SQUASH), and MRP generations. Since none of them are done concurrently, this is the same area. All reports and messages are also done from another area.

The segmentation feature of COBOL is used to reduce the core required for execution. Most distinct functions are in their own segments, and the MRP generation is even done in two different segments.

Mainline program flow is shown in Figure 21. Initialization is followed by one basic loop. A common entry point is used by all routines. At this entry point, more input is read, interpreted, and one of the four basic routines is chosen. As each routine finishes, processing returns to this entry point.

Since input is free-form in nature, it is processed in a compiler-like manner in one routine and stored in a fixed format for all other routines. Such things as missing parameters, trailing blanks, leading zeros, and numeric sign are handled here.

The command routine choses the appropriate subroutine (such as LOAD) and simply branches to it. The option routine handles each possible option including the versatile ODATE.

The transaction routine breaks all transactions into four basic types:

Item master maintenance
Order maintenance
Bill of material maintenance
MRP generation

Common logic, such as look-up routines, is heavily utilized. The MRP generation is split into two phases:

Initialize, clear pegs and need dates, and load requirements.
Process all items by low level code.

The report routine sets up appropriate headings first. If an inquiry is required, the specific data is located and one of several print routines is invoked. If a full report or directory is required, the appropriate file is scanned, items are eliminated based upon directory rules, pointers are sorted, and sorted pointers are used to invoke one of several print routines. Some print routines use others, such as the stock status using the item master and order print routines.

Various general modules are used throughout the program, such as:

- Sort pointers
- Look-up routines
- Pack and unpack dates

The result is a core-stingy, modular, expandable, single-program system. The source code spans more than 4000 cards.

IX. A SAMPLE INVESTIGATION

As an example of the investigative capabilities of this package, a premise has been tested. Master scheduling has a major effect on material requirements. Changes in the schedule are often necessary, but they require many changes throughout the manufacturing facility. Whimsical changes will be costly, and necessary changes should be cautiously implemented. Certain guidelines should be followed when manipulating the master production schedule.

The obvious guideline relates to lead time. Changes should be implemented within the lead time only when deemed absolutely necessary. When they are made within this time, higher costs should be anticipated. Verification of this premise is reasonably simple.

When changes are made beyond the lead time, but still within the "near horizon", dates should be changed rather than quantities. The "near horizon", as used here, is intended to mean beyond lead time but close enough to require some action. This premise is not quite as obvious.

To investigate this, a plan was conceived to simulate perfect compliance with MRP requests followed by two different schedule changes. Specifically, every requisition became an order, every order was rescheduled or cancelled as requested, and the shop order for 150 tricycles was received into stock as follows:

Per Figure 9, add purchase orders for:
- 11 FRM-1 wanted 10/10/77
- 78 HUB-12 wanted 8/15/77

Per Figure 5, reschedule purchase orders:
- PO-12 to 9/19/77
- PO-14 to 9/5/77
- PO-17 to 9/19/77
- PO-25 to 9/12/77

Per Figure 5, cancel purchase orders:
- PO-37
- PO-41
- PO-52

Per Figure 6, reschedule shop orders:
- RWHA25-3 to 8/8/77
- WASSY14-5 to 9/5/77

Per Figure 6, receive:
- 150 tricycles on TRI77-1

This simulated perfect compliance with requirements was verified by causing another generation effective 6/13/77. No requisitions were required, and all orders were needed on their want dates.

The master schedule for TRI-77 was changed in two ways. First, September through December requirements were rescheduled forward ½ month as follows:

TRI77SEP was changed to 8/15/77
TRI77OCT was changed to 9/15/77
TRI77NOV was changed to 10/15/77
TRI77DEC was changed to 11/15/77

Second, August, September, and November requirements were increased by five each as follows:

TRI77AUG was changed to 30
TRI77SEP was changed to 30
TRI77NOV was changed to 35

The first change included date changes only. Requisitioning produced no new stock requirements. Order directories showed rescheduling required on all orders.

The second change included quantity changes only. Requisitioning produced new stock requirements:

PART NUMBER	QTY	WANT
FRM-1	15	10/10/77
HUB-12	45	8/22/77
RWHA-25	6	8/08/77

Order directories showed rescheduling required on most orders. Orders were needed as follows:

ORDER NUMBER	WANT DATE	NEED change 1	NEED change 2
PO-FRM	10/10/77	9/26/77	10/10/77
PO-HUB	8/15/77	7/18/77	8/01/77
PO-12	9/19/77	9/05/77	9/19/77
PO-14	9/05/77	8/22/77	8/01/77
PO-17	9/19/77	9/05/77	8/22/77
PO-25	9/12/77	8/22/77	9/12/77
RWHA25-3	6/25/77	7/25/77	8/08/77
WASSY14-5	9/05/77	8/22/77	8/08/77

Subjective analysis of the results of this investigation support the premise that dates should be changed rather than quantities whenever possible in the "near horizon". Rescheduling was required in both cases. The major difference was in requisitioning. New stock was required in order to satisfy the quantity changes, but none was required to satisfy the date changes. Requisitions require purchasing time and inventory dollar commitment.

It is an intuitive extension that decreases in the master schedule, using quantity changes, would request order cancellations. Offsetting master schedule changes would request new orders and order cancellations if changes were made by quantity. No such changes would be requested if changes were made by date.

New orders and order cancellations not only raise inventory costs and purchasing demands, but they also can create vendor ill-will. Variation of dates buffers vendors from internal schedule changes, but allows shop priorities to be rearranged.

X. SUMMARY AND CONCLUSIONS

A general purpose MRP system was developed and used for an investigation. It will soon be converted for use on the Ingersoll-Rand Company Rock Drill Division computer system in Phillipsburg, New Jersey. There it will be used for training and demonstration.

The investigation in this thesis, which used the system, yielded a useful answer to a scheduling problem. It suggests that quantity changes in the "near horizon" of the master schedule create more problems than date changes.

With proper interfacing between Ingersoll-Rand data files and this program, it is possible that useful information may be acquired in a business environment. Lehigh University may also find it useful for teaching MRP to industrial engineering students.

XI. RECOMMENDATIONS FOR FURTHER STUDY

MRP, unlike EOQ, is not concerned with lead time, cost, and safety stock. It is, however, dependent upon three very basic items:

> Inventory record accuracy is mandatory. Cycle counting, physical security, and automated storerooms all help this.
>
> Bill of material accuracy is mandatory. Legitimate assembly vs enginerring variations, engineering changes, and customized machines are among the problems existing here.
>
> A realistic master schedule may be the most elusive requirement. Very little work has been done to track scheduling problems or to aid in creating schedules.

Several features have not been included in this package, such as:

> Allocation of stock to open shop orders, time phased to minimize premature pulling or staging for assembly.
>
> Firm planned orders to schedule a receipt in the future and also to schedule requirements for components. This can also set specific lot sizes and lead times on projected orders.
>
> Discrete lot-sizing techniques to lot-size planned orders. Least total cost, Wagner-Whiten method, and many others exist.
>
> Capacity requirements planning to test labor demanded for feasibility and to feed shop floor control.
>
> Shop floor job control, with dispatching and work center queue control.
>
> Various miscellaneous features such as

80-20 or ABC analysis (based on Pareto's Law), obsolete and slow-moving inventory (OSMI) analysis, projected shortages, C'bank lock-up stock quantities, etc.

A new method of MRP, known as net change, could be fully or partially implemented. Partial implementation may ignore engineering changes, lead time changes, and master schedule changes, and rely upon an infrequent generation to clean up.

XII. APPENDIX

A. INTRODUCTION

MRP is a completely self-contained system which can be used to learn about materials requirements planning, and can also be used to demonstrate and investigate inventory problems.

Contained within this package are:

A data base management system.
An inventory accounting system --- a method of maintaining stock balances and orders with their balances.
An engineering information system --- a method of maintaining bills of material.
An MRP system --- a method of deciding what and when to order and reschedule.
Several utilities - e.g., storing the data base to a local file.

This program is written in COBOL, using the segmentation feature. It will execute within 40K of the present Lehigh University computer. The segmentation feature requires that the name be COBCODE; that is, it must be attached as COBCODE, or at least executed as COBCODE, but its permanent name is MRP. Certain files are used by MRP:

INPUT - All input, except a complete data base, are done with this file.
OUTPUT - All reports and messages are done to this file.
MRPIN - A data base (created by MRP) may be loaded from this binary file.
MRPOUT - The current data base may be stored to this binary file.
ERRFILE - A file created and maintained by MRP - used for file I/O errors.

B. DATA BASE

GENERAL

The current data base is maintained within core. It may be stored to, or loaded from, local files in its entirety. Upon initiation, MRP has no data base in core; the data base should either be started from scratch or loaded from local file MRPIN.

The data base can best be described using Figure 1. Each item in the figure is logically a file. A calendar file is available for use by the MRP generation. All other files are related to the item master. An item may have purchase orders, shop orders, customer orders, and a master schedule. The product structure describes engineering bills of material by linking assemblies and subassemblies to their components. Pegging details requirements from a higher level.

Within this package, the calendar and item master files are separate tables. Two other tables exist. The order table contains purchase, shop and customer orders, and master schedule items. The chain table contains product structures and pegs.

Table sizes are:

Calendar:	100
Item master:	50
Orders:	125
Chains:	300

FILES

Calendar file: One date for each planning period.

Item master file: One item for each unique part.
Key - part number
Directly maintained - description, lead time
Indirectly maintained - on-hand balance, low level code

Purchase order file: One order for each part and want date.
Key - order number
Directly maintained - quantity, want date
Indirectly maintained - need date

Shop order file: One order for each part and want date.
Key - order number
Directly maintained - quantity, want date
Indirectly maintained - need date

Customer order file: One order for each part and want date.
Key - order number
Directly maintained - quantity, want date

Master schedule file: One entry for each part and want date.
Key - order number
Directly maintained - quantity, want date

Product structure file: One entry for each parent-component.
Directly maintained - quantity

Pegging file: One entry for each parent, component, and want date.
Indirectly maintained - quantity, date

C. INPUT

GENERAL

Input to MRP is basically free-form. Each eighty column input is processed based upon its content, starting from column one until the first blank column. All information found beyond the first blank is considered comments. All input follows the form:

operation(,param1)(,param2)(,param3)(,param4)

All parameters are positional, which means that intervening commas define which parameter is intended (e.g. operation,,param2). Certain characters have special meaning. Commas and blanks may never be part of any alphanumeric or numeric data. All numeric items are assumed positive (plus sign is optional) unless preceded by a minus sign (dash). All alphanumeric fields are limited to ten characters (trailing blanks are not used), and all numeric values (other than dates) may not exceed 4095.

The operations are divided into four basic categories. The first character determines which type of operation it is:

T - transaction - e.g. TAIM
R - report request - e.g. RIMI
O - option - e.g. ONAME
other - command - e.g. LOAD

All input is checked for syntax and its potential effect on the data base. If anything is wrong, a meaningful message is printed (e.g. NEGATIVE ORDER BALANCE, or MUST BE NUMERIC), and the input is ignored.

TRANSACTIONS

Transactions are input which affect data base content at a logical level. All transactions follow the form Taff(,parameters) - where "a" is the action, and "ff" is the file (sometimes not required). The values of "a" and "ff" are:

a	ff
A - add	IM - item master
C - change	PO - purchase order
D - delete	SO - shop order
J - adjust	CO - customer order
R - receive	MS - master schedule
I - issue	BM - bill of material
G - MRP generation	

e.g. TABM,HBAR-2,BAR-4,1
TRPO,PO-12,25

REPORT REQUESTS

Report requests print logical data base information. All requests follow the form Rrrt(,key) - where "rr" is the specific report requested, and "t" is the type of inquiry. Values of "r":

IM - item master information
PO - purchase order information
SO - shop order information
CO - customer order information
MS - master schedule information
RQ - requisitioning information
OS - order status information
SS - stock status information

BM - single-level bill of material
WU - single-level were-used
FB - full (indented) bill of material
FW - full (indented) were-used

Each of these report types may be followed by values of "t":

F - full listing
D - directory (limited listing)
I - inquiry (one specific key)

e.g. RSSD
RFBI,TRI-77

OPTIONS

Options change overall system parameters, miscellaneous information and the calendar file. All options begin with the letter "O", and have parameters. The options are:

OECHO - option to echo all input
ONAME - change data base name
OREQ - change requisitioning horizon
OHORIZ - change planning horizon
ODATE - maintain or print calendar file
OSSREQ - option to print requisitioning information on stock status
OSSMAT - option to print MRP matrix on stock status
OSSORD - option to print orders on stock status
OSSPEG - option to print pegging on stock status

e.g. OECHO,OFF
OREQ,4

COMMANDS

Commands are utilities which initialize the data base, control data base input and output, print options, aid in debugging, and complete execution. They may not begin with T, R, or O. The commands are:

PRINT - print basic data base information and options
SCRATCH - scratch the present data base
LOAD - load the data base from MRPIN
STORE - store the data base to MRPOUT
SQUASH - utility to reorganize the data base
DUMP - utility to debug the data base
DONE - stop the program

D. THE SYSTEM

SAMPLE DATA BASE

A sample data base is available which serves as a good learning tool. This manual will use examples from it. The data base is that of a plant which manufactures tricycles.

The TRI-77 is a very specific tricycle, and it is the only product line they have. The tricycle and its subassemblies are assembled at the plant from purchased parts. A few customer orders exist for the tricycle, and also spare parts demand for the rear wheel assembly. Most demand is derived from a master schedule.

ITEM MASTER

Figure 4 is an item master directory of the sample data base. This report was created by entering "RIMF". Entering "RIMD" would yield a list of all items with on-hand balances; that is, HUB-12 would not appear. Entering "RIMI,FRM-1" would produce a printout of FRM-1 only.

Part number AX-3B describes a specific axle. A similar axle may have a similar part number, but if they are not the same part, the numbers should be unique. The axle has a lead time of seven planning periods (weeks in the sample data base). This means that the time between starting an order (for example, writing a purchase order) and receiving axles into stock, is normally seven weeks. We have 700 axles on hand (in stock). The low level code (LLC) will be described later.

In order to add a part number to the data base, a "TAIM" transaction is needed. This has the form:

```
TAIM,(part number),(description),(lead time)
3.g. TAIM,AX-3B,AXLE,7
```

In order to change the description or the lead time, a "TCIM" transaction is needed. This has the form:

```
TCIM,(part number),(description),(lead time)
```

where only the fields to be changed are used.

```
e.g. TCIM,AX-3B,BOLT      to change description
     TCIM,AX-3B,,15       to change lead time
     TCIM,AX-3B,BOLT,15   to change both
```

In order to delete a part (once it is no longer needed), a "TDIM" transaction is needed. This has the form:

```
TDIM,(part number)
e.g. TDIM,AX-3B
```

The on-hand balance may be directly adjusted using the "TJIM" transaction. This has the form:

```
TJIM,(part number),(quantity)
e.g. TJIM,AX-3B,-500 to reduce on-hand by 500
```

ORDERS

Figure 5 is a purchase order directory of the sample data base. This report was created by entering "RPOF". Entering "RPOD" would yield a list of all purchase orders with open balances; that is, PO-29 would not appear. Entering "RPOI,PO-25" would produce a printout of PO-25 only.

Purchase order PO-12 shows a scheduled receipt for 50 bars of BAR-4 on 6/6/77. The MRP system has decided that this order is not needed until 9/19/77 (the reason for this will be explained later). Note that 150 more bars are scheduled for 9/17/78 (PO-41), and are not needed (need date 99/99/99).

In order to add a purchase order to the data base, a "TAPO" transaction is needed. This has the form:

```
TAPO,(part number),(order number),(quantity),
                                   (want date)
e.g. TAPO,BAR-4,PO-12,50,60677
```

In order to change the quantity or want date, a "TCPO" transaction is needed. This has the form:

```
TCPO,(order number),(quantity),(want date)
```

where only the fields to be changed are used. Note that the part number is not needed.

```
e.g. TCPO,PO-12,40          to change quantity
     TCPO,PO-12,,91977      to change want date
     TCPO,PO-12,40,91977    to change both
```

In order to delete a purchase order that is complete, a "TDPO" transaction is needed. It has the form:

```
TDPO, (order number)
e.g. TDPO,PO-12
```

All orders are handled in the same manner, i.e.

Shop orders - TASO,TCSO,TDSO,RSOF,RSOD,RSOI
Customer orders - TACO,TCCO,TDCO,RCOF,RCOD, RCOI

Master schedule items - TAMS,TCMS,TDMS,
RMSF,RMSD,RMSI

Shop orders and purchase orders are scheduled receipts with changing need dates. Customer orders and master schedule items are demands, which do not have need dates.

Figure 10 shows an order status report for part TRI-77. This report was created by entering "ROSI,TRI-77". Entering "ROSF" would print this information for all parts, and entering "ROSD" would print this information for all parts with orders.

Basic part information is printed first, followed by all purchase orders, shop orders, customer orders, and master schedule items, each within want date sequence. The note in front of the order number (PURC, SHOP, CUST, or MAST) describes the type of order. Need dates are only printed for purchase and shop orders, since they do not apply to others.

RECEIPTS AND ISSUES

Stock is received when it is added to on-hand from any of several sources. This may occur when all or part of a purchase order arrives at the stocking location, when a shop order is either partially or completely finished and the finished part is stocked, or when some unexpected source makes stock available (for example, a transfer from another location).

An unscheduled receipt (no orders involved) merely adds to the on-hand balance of the part. This is done using the "TRIM" transaction of the form:

```
TRIM,(part number),(quantity)
e.g. TRIM,BAR-4,5 would add five to the
                  on-hand balance of BAR-4
```

A scheduled receipt involves either a purchase order or a shop order, and it adds to the on-hand balance of the part while it subtracts from the open balance of the order. This is done (for a purchase order) using the "TRPO" transaction of the form:

```
TRPO,(order number),(quantity)
e.g. TRPO,PO-12,5 would subtract five from
                  the open balance of PO-12
                  and add five to the on-
                  hand balance of BAR-4
```

The "TRSO" transaction functions identically for shop orders.

Received stock may be returned (for example, a shop order rejection) by merely using a negative quantity, such as TRPO,PO-12,-5. No transaction will be allowed to create either a negative open balance or an on-hand balance.

Stock is issued when it is drawn from storage to be used for any of several reasons. This may occur when all or part of an unscheduled demand (for example, testing), or a scheduled demand (customer order or master schedule) is satisfied.

An unscheduled issue (no order involved) merely subtracts from the on-hand balance of the part. This is done using a "TIIM" transaction of the form:

```
TIIM,(order number),(quantity)
e.g. TIIM,TRI-77,5 would reduce the on-hand
                   balance of TRI-77
```

A scheduled issue involves either a customer order or a master schedule item, and it subtracts both from the on-hand balance of the part and the open balance of the order. This is done (for customer orders) using a "TICO" transaction of the form:

```
TICO,(order number),(quantity)
e.g. TICO,ATLANTA5,5 would reduce the order
     balance of ATLANTA5, and the on-hand
     balance of TRI-77 both by five
```

The "TIMS" transaction functions identically for master schedule items.

Issued stock may be returned (for example, a customer rejection) by merely using a negative quantity, such as TICO,ATLANTA5,-5. No transaction will be allowed to create either a negative open balance or an on-hand balance.

BILLS OF MATERIAL

Figure 14 is a single-level bill of material for TRI-77. This report was created by entering "RBMI,TRI-77". Entering "RBMF" would yield this information for all parts, and entering "RBMD" would yield this information for all parts with components.

The first line describes basic item information for TRI-77. Each line below the first describes a component of TRI-77 with the quantity of each used. The tricycle is made of:

```
1 frame (FRM-1)
1 handlebar (HBAR-2)
2 pedals (PED-1)
1 wheel assembly (RWHA-25)
1 seat (SEAT-2A)
1 wheel (WASSY-14)
```

Note that TRI-77 was listed at level zero with a quantity of one, and that all other parts were listed at level one (and indented one position as shown by periods), with their respective quantities.

This bill of material is the result of six product structure records. Each product structure describes a parent-component relationship. These product structure records are added to the data base using a "TABM" transaction of the form.

```
TABM,(parent),(component),(quantity per)
e.g. TABM,TRI-77,FRM-1,1
     TABM,TRI-77,PED-1,2
```

In order to change the quantity in a product structure record, a "TCBM" transaction is used:

```
TCBM,(parent),(component),(quantity per)
e.g. TCBM,TRI-77,FRM-1,3
```

In order to delete a product structure record, a "TDBM" transaction is used:

```
TDBM,(parent),(component)
e.g. TDBM,TRI-77,FRM-1
```

Figure 16 is a multi-level bill of material for TRI-77. This report was created by entering

"RFBI,TRI-77". Entering "RFBF" would yield this information for all parts, and entering "RFBD" would yield this information for all parts which have components and are not components of any other part.

The first line describes TRI-77. Items listed with one dot are at level one, and correspond to the single-level components of TRI-77. HBAR-2 has components of one bar and two grips. These items appear indented below HBAR-2. RWHA-25 has components of one axle and two WASSY-14 wheels. This wheel further has components, and since two wheels are needed, its component quantities are exploded by a quantity of two (doubled). Specifically, RWHA-25 needs one hub, one rim, and one tire. This explosion process shows the total parts required to make one tricycle.

Note the correspondence of level number to the number of dots or the indentation of the part. WASSY-14 is listed two times - once at level two and once at level one. The lowest level (highest level number) that WASSY-14 appears at (relative to level zero) is level two. This is the low level code (LLC) for the part. The use of this low level code will be described later.

The multi-level bill of material would be impossible to print (and requirements for TRI-77 impossible to compute) if, for example, a product structure was erroneously added to show that WASSY-14 has a component of one RWHA-25. This would show that WASSY-14 has a component of one RWHA-25, which has a component of two WASSY-14's, which has one RWHA-25, etc. This is the reason for a thorough check to ensure product structure integrity - TABM,WASSY-14,RWHA-25,1 would produce the message "WASSY-14 USED ON RWHA-25".

A product structure also defines the component-parent relationship (opposite of the bill of material), known as the where-used. Figure 15 shows a single-level where-used for WASSY-14. This was produced by entering "RWUI,WASSY-14". Entering "RWUF" would produce this information for all parts, and entering "RWUD" would produce this information for all parts with usage. This shows that WASSY-14 is used on RWHA-25 and TRI-77.

Figure 17 shows a multi-level (indented) where used for HUB-12. This was produced by entering "RFWI,HUB-12". Entering "RFWF" would produce this information for all parts, and entering "RFWD" would produce this report for all parts with usage and no components. It is analogous to the multi-level bill of material.

CALENDAR

Figure 2 shows the shop calendar used in the sample data base. This was created by entering "ODATE,PRINT,ALL". The planning horizon is 100 periods, which may be days, weeks, months, or years. Since the planning periods are weeks, the 100 dates shown are successive Mondays beginning with 6/6/77. Leap years are considered, and since every fourth century is not a leap year (the year 2000 will not be one), there exists an option to decide whether or not the year 00 is a leap year.

Because of the two digit representation of the year, it is possible that the earliest year in the century may not be 00. Specifically, 6/6/77 comes before 6/6/20 if the actual dates are 6/6/1977 and 6/6/2020. An option exists to determine the year which begins the century. When planning by other than year, the century arbitrarily begins 25 years before the first calendar year by default. When planning by year, the first year must begin the century (since the planning period covers 100 years). This is shown by the example where the horizon begins on 6/6/77, so the century begins in 52. This default may be overridden unless planning by year.

The "ODATE" function can be used to print, set parameters, and generate the 100 dates. To print parameters, enter "ODATE,PRINT". This would yield the date, horizon, status (generated or not), century beginning year, and whether or not the year 00 is a leap year. Entering "ODATE,PRINT,ALL" would yield this information and a list of the 100 dates (if generated).

Date parameters may be set using the form:

```
ODATE,(date),(century begin),(LEAP)
                             (NO)
```

using only the parameters to change.

e.g. ODATE,60677 to change the year to 6/6/77
ODATE,,57 to change the century begin year to 57
ODATE,,,LEAP to make 00 a leap year
ODATE,60677,,NO to change year and make 00 not a leap year

The 100 dates may be generated by entering "ODATE,GEN". This will automatically change the default century beginning year. Entering "ODATE,GEN,NO" will generate the dates without changing the century beginning year.

TIME PHASED ORDER POINT

Normal order point systems assume constant, or continuous demand, and therefore determine the need for an order point based on demand over lead time plus safety stock. Time phased order point assumes discrete demand, simulates usage (which includes perfect rescheduling of orders), determines when stock-outs should occur, and plans orders in time to cover the potential stock-outs.

Figure 12 is a stock status for TRI-77. This will show the effect of time phased order point. This was created by entering "RSSI,TRI-77". Entering "RSSF" would produce this information for all parts, and entering "RSSD" would yield this information for all parts with requirements or orders.

The first line describes the basic item master information for TRI-77. The next section is the MRP matrix. This is followed by direct demand (customer orders and master schedule) and the scheduled receipts (purchase and shop orders). The MRP matrix is a graphic representation of simulated stock usage over time.

The matrix only shows dates that have different stock situations. Dates correspond to shop calendar dates (see Figure 2), so items which relate to dates before 6/6/77 appear on 6/6/77, and items which relate to dates after 4/30/79 are dropped.

The first column in the matrix shows the date to which the row relates. The next column shows gross requirements (pure demand). In this case, all requirements are enumerated in the list of direct demand. Any item wanted between calendar dates are shown due on the preceeding calendar date. This is shown by customer order ATLANTA5 wanted on 6/12/77. Since it is needed between 6/6/77 and 6/13/77, the gross requirement for 35 is shown on 6/6/77.

Two pairs of columns follow gross requirements. SCHED shows all scheduled receipts by want date, and NEED shows all needed scheduled receipts by need date. Each is followed by a column called AVAIL which shows the available stock balance if orders are received on want and need dates respectively. In the cited sample, an on hand balance of five tricycles exists, 35 are needed on 6/6/77, and an order is due for 150 on 6/6/77. This leaves 120 in stock on 6/6/77, that is, on hand (5) + scheduled receipts

(150) - demand (35) = available (120). As requirements appear, the available balance decreases and eventually goes negative.

On 9/26/77, a negative available balance (stock out) of 11 tricycles will occur unless stock is received. This means we have a net requirement for 11 tricycles on 9/26/77. On 10/31/77, we have a net requirement for 30 more. Due to the seven week lead time (L/T = 7), we should plan orders seven weeks before the net requirements. Seven weeks before 9/26/77 is 8/8/77. This means we should initiate an order for 11 tricycles on 8/8/77 to get them by 9/26/77 and avert a stock-out. The columns NET and PLAN show net requirements and planned orders. Note that no lot-sizing is used, although it would be a straightforward extension to lot-size planned orders, using any of several widely accepted techniques for discrete lot-sizing.

MRP

Time phased order point works well for items without dependent demand (demand generated because of projected assembly of other items). The components of TRI-77 will derive some demand from the planned orders of the tricycle.

Figure 11 is a stock status for WASSY-14. This will show the effect of dependent demand on an item. WASSY-14 is the wheel which is used directly on the tricycle (the front wheel), and also used on the rear wheel assembly (RWHA-25). Customer orders SP-ATL-12 and SP-SEA-17 show direct spare parts demand on the wheel.

Pegged demand lists all demand "exploded" from a higher level. The entries with part number TRI-77 show the quantities and dates of planned orders of TRI-77 (see Figure 12). RWHA-25 also has planned orders, and the bill of material shows two wheels are needed for one RWHA-25; so a pegged requirement for 52 wheels exists for a planned order of 26 RWHA-25's.

Both direct demand and pegged demand are combined into gross requirements. Note 8/8/77 has a requirement of 65 from customer order SP-SEA-17, and pegged requirements from RWHA-25 and TRI-77.

Note that shop order WASSY14-5 is wanted on 6/20/77, but that would carry inventory for a long time that is not needed (see SCHED-AVAIL). This order is not needed until 9/5/77, as shown in the NEED--AVAIL columns. This is the reason that the need date has been set to 9/5/77.

REQUISITIONING

Figure 13 shows the stock status for HUB-12. This has a planned order on 6/6/77, and another on 6/27/77. Assuming the policy is to create orders for all planned orders covering the next four weeks, we should create an order for this item. To create an order, we need a quantity and a want date. The line above the MRP matrix for HUB-12 shows a requisition for 78 items wanted on 8/15/77.

Requisitioning requires a horizon (4 weeks in the sample data base) to create orders corresponding to planned orders. All orders planned within that four week period are summed to obtain the requisition quantity (in the example, 43 + 35 = 78). This requisition is offset from the first requirement date by the lead time (in the example, 10 weeks + 6/6/77 = 8/15/77). This date is the original want date of the requisition.

Figure 9 shows a requisitioning directory for the sample data base. This was created by entering "RRQD". Only parts with a requisition are shown. Entering "RRQF" would show requisitioning for all parts, and entering "RRQI,FRM-1" would show requisitioning for FRM-1 only. The directory should be used to review all requisitions and create orders. This is the responsibility of the user.

GENERATION

As items and orders are added, changed, and deleted, stock received and issued, and engineering changes made, simulated stock usage changes. Net change MRP would automatically recalculate planned orders and need dates, but this is time-consuming, inefficient, difficult to implement, and often not necessary.

This package is regenerative rather than net change; that is, MRP information is created during an MRP generation, and not updated until the next generation. Each generation "throws out" the old MRP information and creates the new information from scratch. The generation is initiated using the "TG" transaction of the form:

TG(,,ALL)

where "ALL" is used only if exception messages are desired.

Need dates of all new orders are added as 00/00/00. An MRP generation will set need dates on all needed orders which have an open balance, and will set the need date to 99/99/99 on all unneeded orders.

Once an MRP generation occurs, any further maintenance to the data base makes the information less accurate. It is necessary to set a beginning date and generate a calendar before running the generation. In order to print the stock status or requisitioning, the calendar file must be identical to its status at last generation.

The generation is done by simulating each part in low level code sequence starting from level zero. This ensures that an item has all requirements available before it is processed. For example, WASSY-14 cannot be simulated until after all planned orders are known for both TRI-77 and RWHA-25.

E. MESSAGES

Many messages are printed by MRP. Some messages are comments, some are warnings, and other cite attempted errors. This package is designed for data integrity whenever possible; that is, it is difficult to do things which are illogical, such as issuing more stock than available.

Upon initiation of the package, the message "START MRP" will appear. Throughout execution of the program, the message "ENTER" will appear whenever input is required. When execution ends (by entering "DONE" or by end-of-file on INPUT), the message "DONE MRP" appears.

Certain messages relate to data base content. When MRP is initiated, no data base exists. It is expected that the next command will be either "SCRATCH" or "LOAD". If anything else is attempted, the message "NO DATA BASE" will appear. If any files are filled, messages are shown, such as "ITEM FILE FULL".

All input to MRP may be categorized as commands, options, transactions, and report requests. The first part of any input must be a valid function, so if it is not recognized, a message such as "xxxxxxxxxx BAD COMMAND" will appear, where xxxxxxxxxx is the first part of the input.

Once the function is recognizable, certain syntax must be followed. Many messages describe syntax errors, such as:

```
MUST BE NUMERIC
MUST BE 1-100
MUST BE ON OR OFF
BAD QUANTITY
BAD DATE
NO NAME SPECIFIED
MUST BE A VALID PERIOD
NO ORDER NUMBER
NO PART NUMBER
NO PARENT
NO COMPONENT
MUST HAVE A DESCRIPTION
LEAD TIME OUT OF RANGE
NOTHING TO CHANGE
xxxxxxxxxx BAD QTY - xxxxxxxxxx is
                     the field
xxxxxxxxxxBAD LEAD TIME - where xxxxxxxxxx
                          is the field
```

During file maintenance, data may not be added twice, and certain data may not exist on file when needed. These circumstances cause messages like:

```
xxxxxxxxxx DUPLICATE - where xxxxxxxxxx is
                       is the key
PRODUCT STRUCTURE ON FILE
PRODUCT STRUCTURE NOT ON FILE
```

When deleting data, certain checks are made to determine inactivity. If any errors exist, messages appear like:

```
PART STILL HAS DETAIL RECORDS
PART HAS AN ON HAND BALANCE
OPEN BALANCE EXISTS
```

Order balances and on-hand balances are altered using various transactions. They must never go below zero, nor above 4095. When these events would occur, messages are printed like:

```
NEGATIVE ORDER BALANCE
ON HAND BALANCE EXCEEDS MAXIMUM
MAXIMUM SUBSTITUTED
```

Various functions may require very specialized messages, such as:

pppppppppp IS USED ON cccccccccc
where pppppppppp is the parent part and cccccccccc is the component part of a bill of material add transaction. It is found that the parent is already used on the component. This would cause problems if allowed to function.

GENERATE DATES
an MRP generation is requested, but the calendar file is not generated.

CALENDAR AND GEN OUT OF SEQUENCE
a stock status or requisitioning report cannot be done correctly, because the calendar file has changed since the last MRP GENERATION.

When using the "LOAD" command, the assumption is made that local file MRPIN contains an MRP data base. If this is not the case, one of the following messages may appear:

NO INPUT
NOT AN MRP DATA BASE
NO CONTROL INFORMATION
ERROR IN LOAD, FILE SCRATCHED
TOO MANY RECORDS

If any of these messages appear, MRPIN is not a data base created using the "STORE" command of MRP.

An MRP generation is initiated using the "TG" transaction. During this generation, many exceptions may occur. If these exceptions are to be printed, the generation may be initiated by entering "TG,,ALL". The following messages may be printed using this option of "TG":

xxxxxxxxxx CO= cccccccccc EXCEEDS HORIZON
Part number xxxxxxxxxx has a customer order cccccccccc that is wanted beyond

the 100th calendar date - it is not considered as a requirement. The same may happen to a master schedule item.

xxxxxxxxxx GROSS REQUIREMENT TRUNCATION

Part number xxxxxxxxxx has a gross requirement in excess of 4095, 4095 is subsitituted. This may also happen to planned orders.

xxxxxxxxxx TOO MANY REQUIREMENTS

Part number xxxxxxxxxx has more than 18 unique requirements, the ones which are furthest into the future are dropped. This may also occur with planned orders.

pppppppppp - cccccccccc REQUIREMENTS EXCEEDS MAXIMUM

Parent part pppppppppp planned order exploded through the bill of material to component cccccccccc gross requirement exceeds 4095, 4095 is substituted.

F. SYNTAX

COMMANDS

PRINT - Will print basic control information (see Fig 3).

SCRATCH - Will create an initialized data base.

LOAD - Will scratch the present data base, and load a new data base from local file MRPIN. This file must have been created using the "STORE" command.

STORE - Will create the file MRPOUT with the contents of the present data base after reorganization (see SQUASH). This file may subsequently be changed to MRPIN and LOAD'ed into MRP. Note that this is a binary file (not coded).

DONE - Will end execution of MRP.

SQUASH - A utility to reorganize the data base to move deleted space beyond all good space. This is automatically done during the STORE command. It has no noticeable effect on program efficiency; it merely makes less data necessary on a STORE. This has no common use to the MRP user.

DUMP - A utility of no actual use to the MRP user. It displays crude file for debugging the data base logic. The options are:

```
DUMP - prints all good parts, orders, and
       chains.
DUMP,ALL - prints deleted space as well.
DUMP,PART,(ALL) - prints all good parts or
                  only one part.
DUMP,ORDER,(ALL) - prints all good orders or
           (type)  only orders of one type
                   (e.g. P-purchase)
DUMP,CHAIN,(ALL) - prints all good chains,
          (type)  or only chains of one
                  type (e.g. B-bills)
```

OPTIONS

ONAME,xxxxxxxxxx - Will change the internal name (printed) of the data base to xxxxxxxxxx.

OECHO,(ON) - Will cause all following input to be echoed
(OFF) (if ON), or not echoed (if OFF).

OSSREQ,(ON) - Stock status will print requisitions (if
(OFF) any exist) if this option is on.

OSSMAT,(ON) - Stock status will print the MRP matrix (if
(OFF) any) if this option is on.

OSSORD,(ON) - Stock status will print orders (if any) if
(OFF) this option is on.

OSSPEG,(ON) - Stock status will print pegged requirements
(OFF) (if any) if this option is on.

OHORIZ,(period) - Will set the planning periods to DAY, WEEK, MONTH, OR YEAR.

OREQ, (periods) - Will set the number of planning periods to use for requisitioning.

ODATE,PRINT(,ALL) - Will print date parameters and calendar file status. With the "ALL" option, it will also print all 100 generated dates.

ODATE(,date)(,century beginning year)(,LEAP) - Will set the
(NO) calendar
date, the century beginning year, and will make the year 00 either a leap year or not one.

ODATE,GEN(,NO) - Will generate the calendar file based on the date parameters. Will reset the century beginning year unless "NO" is entered.

TRANSACTIONS

TAIM,(part number),(description),(lead time)
Add item master - all fields required.

TCIM,(part number),(description),(lead time)
Change item master description and/or lead time.

TDIM,(part number)
Delete item master.

TAPO,(part number),(order number),(quantity),(want date)
Add purchase order - all fields required.

TCPO,(order number),(quantity),(want date)
Change purchase order quantity and/or want date.

TDPO,(order number)
Delete purchase order.

TASO,(part number),(order number),(quantity),(want date)
Add shop order - all fields required.

TCSO,(order number),(quantity),(want date)
Change shop order quantity and/or want date.

TDSO,(order number)
Delete shop order.

TACO,(part number),(order number),(quantity),(want date)
Add customer order - all fields required.

TCCO,(order number),(quantity),(want date)
Change customer order quantity and/or want date.

TDCO,(order number)
Delete customer order.

TAMS,(part number),(order number),(quantity),(want date)
Add master schedule item - all fields required.

TCMS, (order number),(quantity),(want date)
Change master schedule item quantity and/or want date.

TDMS,(order number)
Delete master schedule item.

TABM,(parent part),(component part),(quantity)
Add product structure record - all fields required.

TCBM,(parent part),(component part),(quantity)
Change product structure quantity.

TDBM,(parent part),(component part),(quantity)
Delete product structure

TJIM,(part number),(quantity)
Adjust item master on-hand balance.

TRPO,(order number),(quantity)
Receive stock from a purchase order.

TRSO,(order number),(quantity)
Receive stock from a shop order.

TRIM,(part number),(quantity)
Receive stock without an order.

TICO,(order number),(quantity)
Issue stock to a customer order

TIMS,(order number),(quantity)
Issue stock to a master schedule item.

TIIM,(part number),(quantity)
Issue stock without an order.

TG(,,ALL) - Perform an MRP generation. The "ALL" option causes all exceptions to print.

REPORT REQUESTS

All reports follow one of the following formats:

RrrF - Print all data in the data base (full)
RrrD - Print only some of the data (directory)
RrrI,key - Print only data identified by key.

The value of rr and the directory contents are listed:

rr	REPORT	DIRECTORY CONTENT
IM	Item master	Need on-hand balance
PO	Purchase order	Need open balance
SO	Shop order	Need open balance
CO	Customer order	Need open balance
MS	Master schedule	Needs open balance
RQ	Requisitioning	Needs requisition quantity
OS	Order status	Needs orders
SS	Stock status	Needs orders or requirements
BM	Single-level bill of material	Need components
WU	Single-level where-used	Need usage

rr	REPORT	DIRECTORY CONTENT
FB	Multi-level bill of material	Need components and no usage
FW	Multi-level where-used	Need usage and no components

Figure 1 - Data Base Diagram

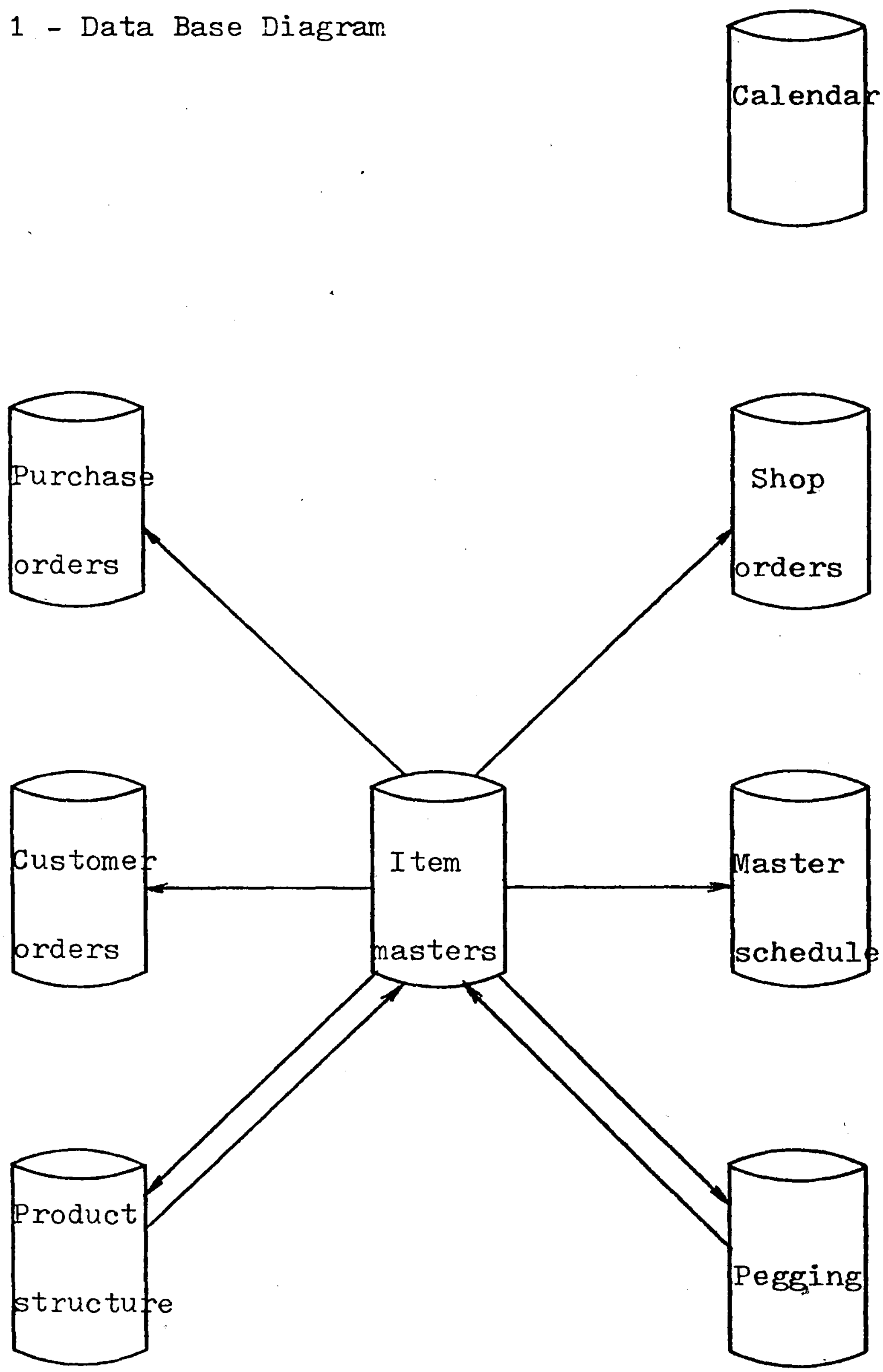


Figure 2 - ODATE report

DATE - 6/06/77, HORIZ-WEEK GENERATED
CENTURY BEGINS 52
YEAR 00 IS NOT A LEAP YEAR

6/06/77	9/19/77	1/02/78	4/17/78	7/31/78	11/13/78	2/26/79
6/13/77	9/26/77	1/09/78	4/24/78	8/07/78	11/20/78	3/05/79
6/20/77	10/03/77	1/16/78	5/01/78	8/14/78	11/27/78	3/12/79
6/27/77	10/10/77	1/23/78	5/08/78	8/21/78	12/04/78	3/19/79
7/04/77	10/17/77	1/30/78	5/15/78	8/28/78	12/11/78	3/26/79
7/11/77	10/24/77	2/06/78	5/22/78	9/04/78	12/18/78	4/02/79
7/18/77	10/31/77	2/13/78	5/29/78	9/11/78	12/25/78	4/09/79
7/25/77	11/07/77	2/20/78	6/05/78	9/18/78	1/01/79	4/16/79
8/01/77	11/14/77	2/27/78	6/12/78	9/25/78	1/08/79	4/23/79
8/08/77	11/21/77	3/06/78	6/19/78	10/02/78	1/15/79	4/30/79
8/15/77	11/28/77	3/13/78	6/26/78	10/09/78	1/22/79	
8/22/77	12/05/77	3/20/78	7/03/78	10/16/78	1/29/79	
8/29/77	12/12/77	3/27/78	7/10/78	10/23/78	2/05/79	
9/05/77	12/19/77	4/03/78	7/17/78	10/30/78	2/12/79	
9/12/77	12/26/77	4/10/78	7/24/78	11/06/78	2/19/79	

Figure 3 - PRINT report

NAME	CREATED	UPDATED	LAST GEN	NEXT GEN
SCRATCH	7/26/77	7/26/77	6/06/77	6/06/77

IM	P/O	S/O	C/O	M/S	B/M	ALL	PEGS
13	8	3	5	6	13	0	33

REQ	RSPCT	PD	IN	OUT	PD	IN	OUT	PD	IN	OUT	PD	IN	OUT	PD	IN	OUT
4	50	2	1	1	4	2	2	10	4	4	20	8	8	50	99	99

ECHO	REQ	RES	MAT	ORD	PEG	HOR
OFF	ON	ON	ON	ON	ON	WK

Figure 4 - Item Masters

RIMF - ITEM MASTER

PART NUMBER	DESCRIPTION	L/T	O/H	LLC
AX-3B	AXLE	7	700	2
BAR-4	BAR	12	10	2
FRM-1	FRAME	15	65	1
HBAR-2	HANDLEBAR	3	50	1
HGRP-14	GRIP	3	10	2
HUB-12	HUB	10	0	3
PED-1	PEDAL	10	50	1
RM-25X2	RIM	7	150	3
RUB-12X1/2	TIRE	3	50	3
RWHA-25	WHEELASSY	5	5	1
SEAT-2A	SEAT	7	150	1
TRI-77	TRICYCLE	7	5	0
WASSY-14	WHEEL	5	75	2

Figure 5 - Purchase Orders

RPOF - PURCHASE ORDER

PURC ORDER	QTY	WANT	NEED	PART NUMBER
PO-12	50	6/06/77	9/19/77	BAR-4
PO-14	100	6/14/77	9/05/77	RUB-12X1/2
PO-17	500	6/28/77	9/19/77	HGRP-14
PO-25	400	8/01/77	9/12/77	PED-1
PO-29	0	8/14/77	0/00/00	HUB-12
PO-37	20	9/16/77	99/99/99	SEAT-2A
PO-41	150	9/17/78	99/99/99	BAR-4
PO-52	500	10/01/77	99/99/99	HGRP-14

Figure 6 - Shop Orders

RSOF - SHOP ORDER

SHOP ORDER	QTY	WANT	NEED	PART NUMBER
RWHA25-3	10	6/25/77	8/08/77	RWHA-25
TRI77-1	150	6/06/77	6/06/77	TRI-77
WASSY14-5	50	6/20/77	9/05/77	WASSY-14

Figure 7 - Customer Orders

RCOF - CUSTOMER ORDER

CUST ORDER	QTY	WANT	PART NUMBER
ATLANTA5	35	6/12/77	TRI-77
CHICAGO7	14	6/18/77	TRI-77
SEATTLE14	12	6/25/77	TRI-77
SP-ATL-12	3	6/13/77	WASSY-14
SP-SEA-17	2	8/09/77	WASSY-14

Figure 8 - Master Schedule

RMSF - MASTER SCHEDULE

MASTER SCH	QTY	WANT	PART NUMBER
TRI77AUG	25	8/01/77	TRI-77
TRI77DEC	35	12/01/77	TRI-77
TRI77JUL	25	7/01/77	TRI-77
TRI77NOV	30	11/01/77	TRI-77
TRI77OCT	30	10/01/77	TRI-77
TRI77SEP	25	9/01/77	TRI-77

Figure 9 - Requisitioning

```
RRQD - REQUISITIONING
-----------------------------------------------------
PART NUMBER  QTY    WANT
FRM-1         11  10/10/77 PDS= 4
HUB-12        78   8/15/77
-----------------------------------------------------
```

Figure 10 - Order Status

```
ROSI - ORDER STATUS
-----------------------------------------------------
PART NUMBER           DESCRIPTION  L/T O/H  LLC
TRI-77                TRICYCLE       7   5    0

     ORDER           QTY    WANT       NEED
SHOP TRI77-1         150   6/06/77    6/06/77
CUST ATLANTA5         35   6/12/77
CUST CHICAGO7         14   6/18/77
CUST SEATTLE14        12   6/25/77
MAST TRI77JUL         25   7/01/77
MAST TRI77AUG         25   8/01/77
MAST TRI77SEP         25   9/01/77
MAST TRI77OCT         30  10/01/77
MAST TRI77NOV         30  11/01/77
MAST TRI77DEC         35  12/01/77
-----------------------------------------------------
```

Figure 11 - Stock Status of WASSY-14

RSSI - STOCK STATUS

PART NUMBER	DESCRIPTION	L/T	O/H	LLC
WASSY-14	WHEEL	5	75	2

	GROSS	SCHED	-AVAIL	NEED-	-AVAIL	NET	PLAN
6/13/77	3	0	72	0	72	0	0
6/20/77	0	50	122	0	72	0	0
8/01/77	0	0	122	0	72	0	13
8/08/77	65	0	57	0	7	0	30
9/05/77	70	0	13-	50	13-	13	35
9/12/77	30	0	43-	0	43-	30	0
10/10/77	35	0	78-	0	78-	35	0

DIRECT DEMAND

ORDER	QTY	WANT	NEED
CUST SP-ATL-12	3	6/13/77	
CUST SP-SEA-17	2	8/09/77	

PEGGED DEMAND

PART NUMBER	QTY	DATE
RWHA-25	52	8/08/77
TRI-77	11	8/08/77
RWHA-25	70	9/05/77
TRI-77	30	9/12/77
TRI-77	35	10/10/77

SCHEDULED RECEIPTS

ORDER	QTY	WANT	NEED
SHOP WASSY14-5	50	6/20/77	9/05/77

Figure 12 - Stock Status of TRI-77

RSSI - STOCK STATUS

PART NUMBER	DESCRIPTION	L/T	O/H	LLC
TRI-77	TRICYCLE	7	5	0

	GROSS	SCHED-AVAIL		NEED--AVAIL		NET	PLAN
6/06/77	35	150	120	150	120	0	0
6/13/77	14	0	106	0	106	0	0
6/20/77	12	0	94	0	94	0	0
6/27/77	25	0	69	0	69	0	0
8/01/77	25	0	44	0	44	0	0
8/08/77	0	0	44	0	44	0	11
8/29/77	25	0	19	0	19	0	0
9/12/77	0	0	19	0	19	0	30
9/26/77	30	0	11-	0	11-	11	0
10/10/77	0	0	11-	0	11-	0	35
10/31/77	30	0	41-	0	41-	30	0
11/28/77	35	0	76-	0	76-	35	0

DIRECT DEMAND

	ORDER	QTY	WANT	NEED
CUST	ATLANTA5	35	6/12/77	
CUST	CHICAGO7	14	6/18/77	
CUST	SEATTLE14	12	6/25/77	
MAST	TRI77JUL	25	7/01/77	
MAST	TRI77AUG	25	8/01/77	
MAST	TRI77SEP	25	9/01/77	
MAST	TRI77OCT	30	10/01/77	
MAST	TRI77NOV	30	11/01/77	
MAST	TRI77DEC	35	12/01/77	

SCHEDULED RECEIPTS

	ORDER	QTY	WANT	NEED
SHOP	TRI77-1	150	6/06/77	6/06/77

Figure 13 - Stock Status of HUB-12

RSSI - STOCK STATUS

PART NUMBER	DESCRIPTION	L/T	O/H	LLC
HUB-12	HUB	10	0	3

REQUISITION 78 8/15/77 PDS= 4

	GROSS	SCHED-AVAIL		NEED--AVAIL		NET	PLAN
6/06/77	0	0	0	0	0	0	43
6/27/77	0	0	0	0	0	0	35
8/01/77	13	0	13-	0	13-	13	0
8/08/77	30	0	43-	0	43-	30	0
9/05/77	35	0	78-	0	78-	35	0

PEGGED DEMAND

PART NUMBER	QTY	DATE
WASSY-14	13	8/01/77
WASSY-14	30	8/08/77
WASSY-14	35	9/05/77

SCHEDULED RECEIPTS

ORDER	QTY	WANT	NEED
PURC PO-29	0	8/14/77	0/00/00

Figure 14 - Single Level Bill of Material

RBMI - SINGLE-LEVEL BILL OF MATERIAL

PART NUMBER	DESCRIPTION	L/T	O/H	LLC	QTY	LVL
TRI-77	TRICYCLE	7	5	0	1	0
.FRM-1	FRAME	15	65	1	1	1
.HBAR-2	HANDLEBAR	3	50	1	1	1
.PED-1	PEDAL	10	50	1	2	1
.RWHA-25	WHEELASSY	5	5	1	1	1
.SEAT-2A	SEAT	7	150	1	1	1
.WASSY-14	WHEEL	5	75	2	1	1

Figure 15 - Single Level Where-used

RWUI - SINGLE-LEVEL WHERE-USED

PART NUMBER	DESCRIPTION	L/T	O/H	LLC	QTY	LVL
WASSY-14	WHEEL	5	75	2	1	0
.RWHA-25	WHEELASSY	5	5	1	2	1
.TRI-77	TRICYCLE	7	5	0	1	1

Figure 16 - Multi-level Bill of Material

RFBI - MULTI-LEVEL BILL OF MATERIAL

PART NUMBER	DESCRIPTION	L/T	O/H	LLC	QTY	LVL
TRI-77	TRICYCLE	7	5	0	1	0
.FRM-1	FRAME	15	65	1	1	1
.HBAR-2	HANDLEBAR	3	50	1	1	1
..BAR-4	BAR	12	10	2	1	2
..HGRP-14	GRIP	3	10	2	2	2
.PED-1	PEDAL	10	50	1	2	1
.RWHA-25	WHEELASSY	5	5	1	1	1
..AX-3B	AXLE	7	700	2	1	2
..WASSY-14	WHEEL	5	75	2	2	2
...HUB-12	HUB	10	0	3	2	3
...RM25X2	RIM	7	150	3	2	3
...RUB-12X1/2	TIRE	3	50	3	2	3
.SEAT-2A	SEAT	7	150	1	1	1
.WASSY-14	WHEEL	5	75	2	1	1
..HUB-12	HUB	10	0	3	1	2
..RM-25X2	RIM	7	150	3	1	2
..RUB-12X1/2	TIRE	3	50	3	1	2

Figure 17 - Multi-level Where-used

RFWI - MULTI-LEVEL WHERE-USED

PART NUMBER	DESCRIPTION	L/T	O/H	LLC	QTY	LVL
HUB-12	HUB	10	0	3	1	0
.WASSY-14	WHEEL	5	75	2	1	1
..RWHA-25	WHEELASSY	5	5	1	2	2
...TRI-77	TRICYCLE	7	5	0	2	3
..TRI-77	TRICYCLE	7	5	0	1	2

Figure 18 - Theoretical Inventory

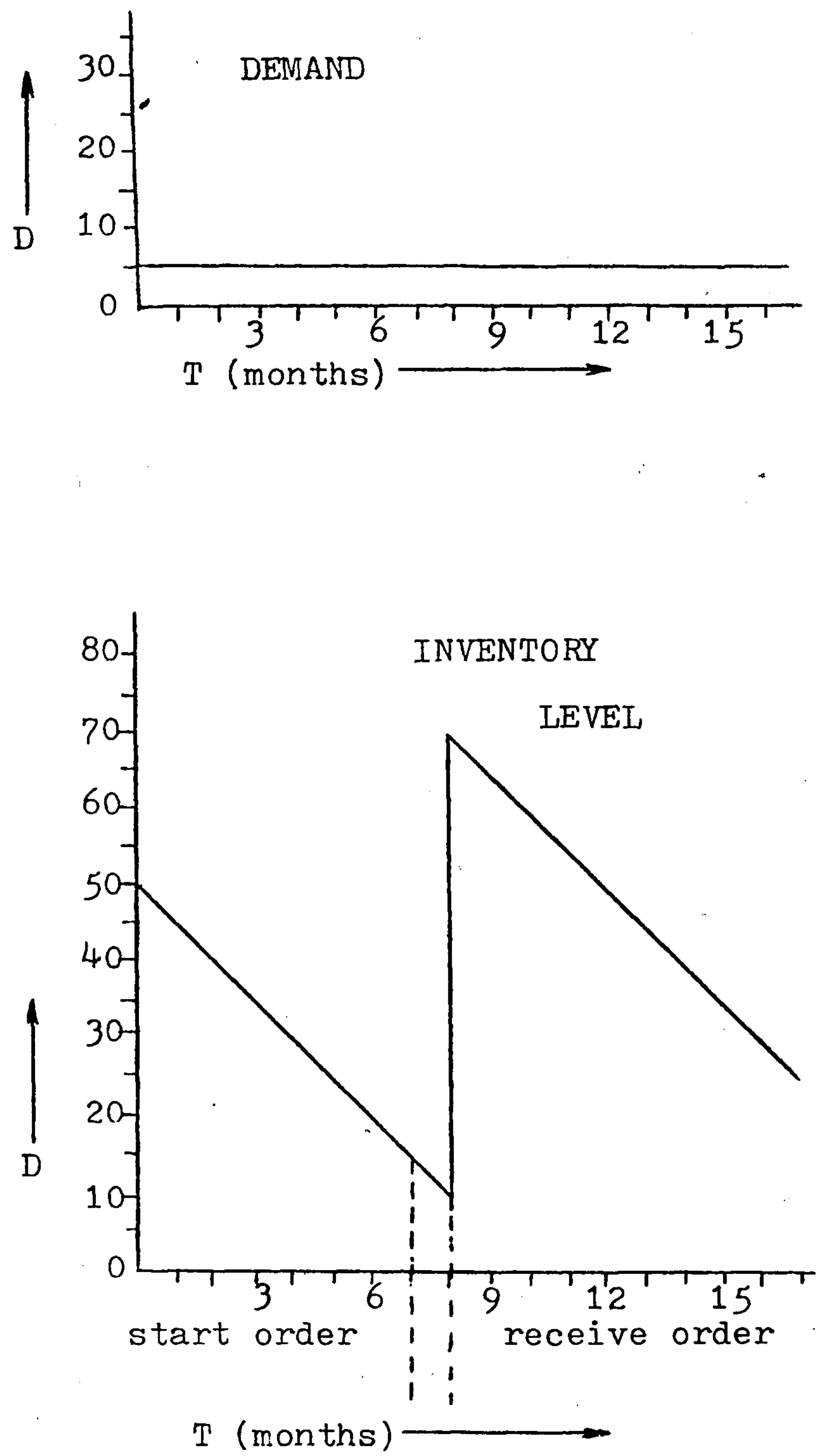
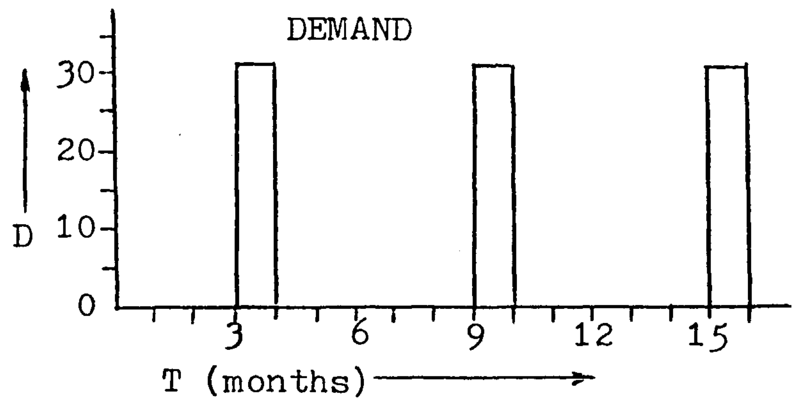


Figure 19 - Actual Inventory



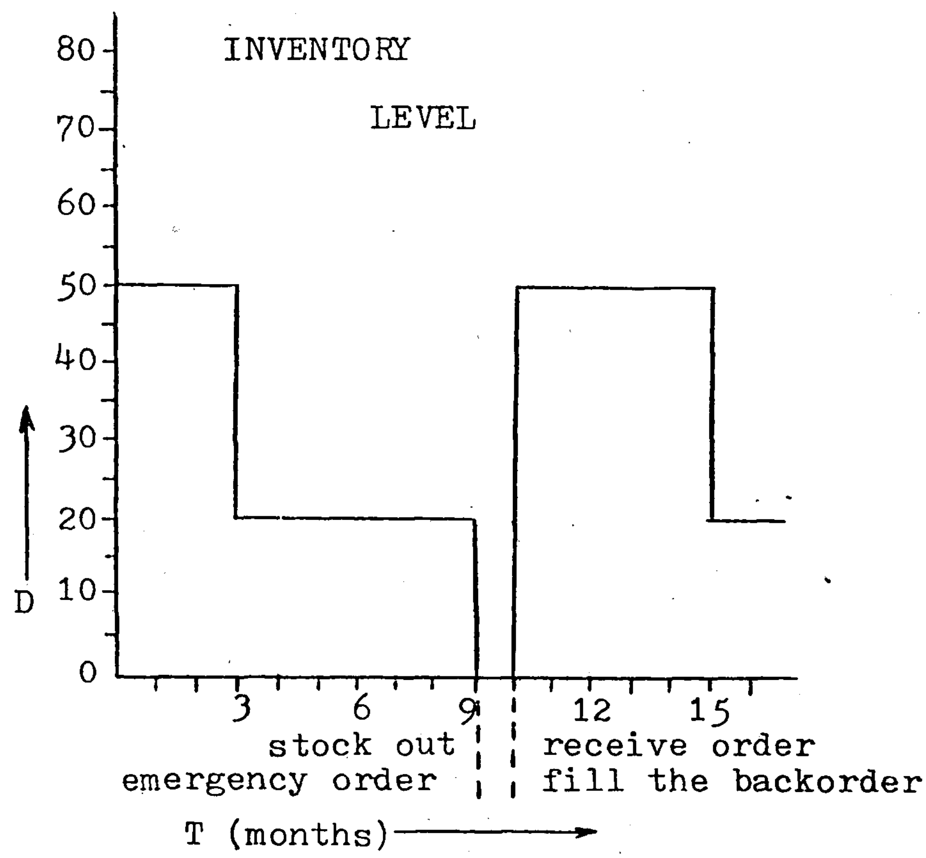


Figure 20 - File Relationships

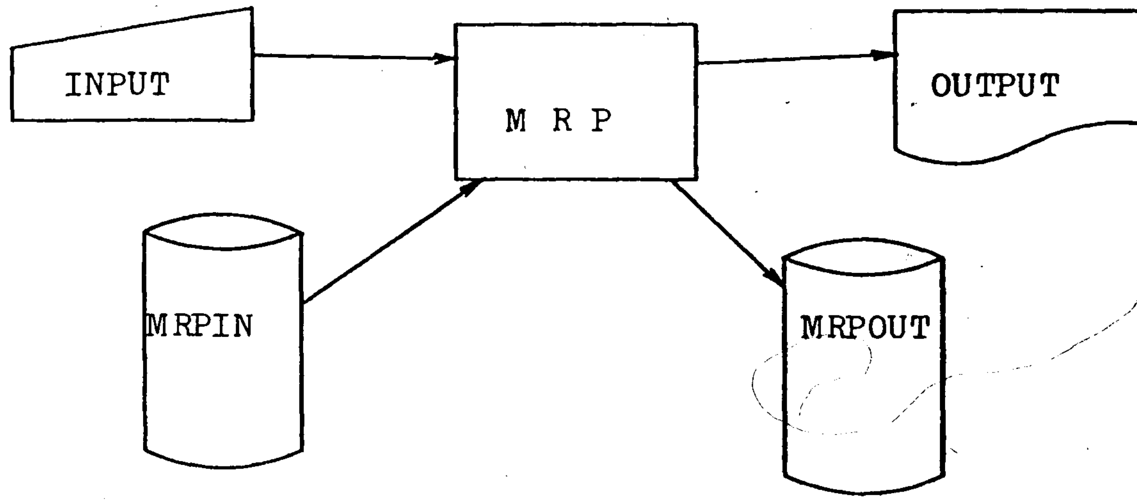
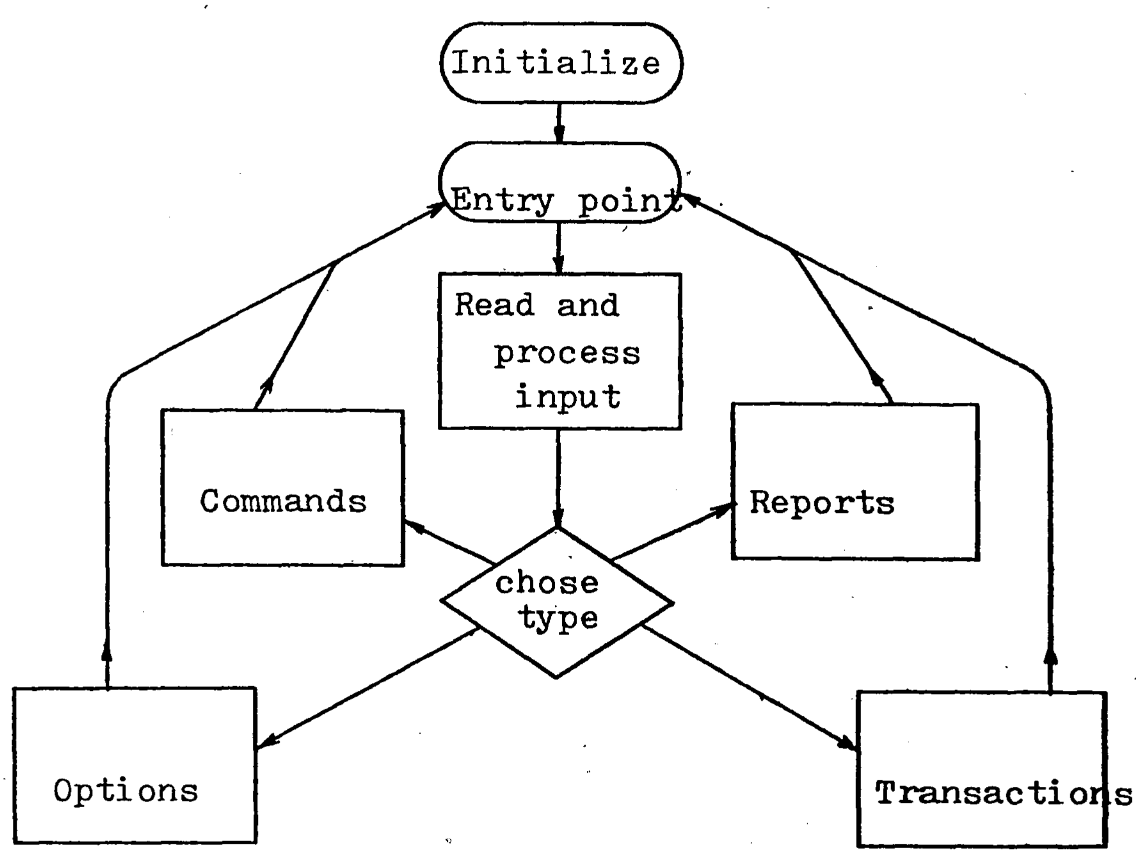


Figure 21 - Program Maintenance



BIBLIOGRAPHY

1. Orlicky, Joseph, Material Requirements Planning, New York: McGraw-Hill Book Company, 1975.

2. Wight, Oliver W., Production and Inventory Management in the Computer Age, Boston: Cahners Books, 1974.

3. Harrington, Joseph Jr., Computer Integrated Manufacturing, New York: Industrial Press Inc., 1973.

4. Plane, Donald R. and Kochenberger, Gary A., Operations Research for Managerial Decisions, Homewood, Illinois: Richard D. Irwin, Inc., 1972.

5. Johnson, Lynwood A. and Montgomery, Douglas C., Operations Research in Production Planning, Scheduling, and Inventory Control, New York: John Wiley & Sons, Inc., 1974.

6. Ingersoll-Rand Company, Rock Drill Division, Phillipsburg, New Jersey, Mark II MRP system.

7. Ingersoll-Rand Company, Rock Drill Division, Phillipsburg, New Jersey, Clarksburg MRP system.

8. Weller, divisionof Cooper Industries, formerly of Easton, Pennsylvania, Burroughs RPS system.

9. Mack Trucks, Inc., Allentown, Pennsylvania, manufacturing system.

V I T A

NAME - Mark Pryor

BORN - JANUARY 2, 1949, Lock Haven, Pennsylvania

PARENTS - R. Louis and Dolores Pryor

WIFE - Diane Pryor

CHILDREN - Scott and Neil Pryor

EDUCATION - Pennsylvania State University
BS Computer Science 1973

Lehigh University
candidate for MSIE

WORK EXPERIENCE - U. S. Navy
Aviation Electronics Technician 4 yrs.

Mack Trucks, Inc.
Programmer 1 yr.

Weller, div. Cooper Industries
Analyst/programmer 1½ yr.

Ingersoll-Rand Company
Programmer/Analyst 2½ yrs.