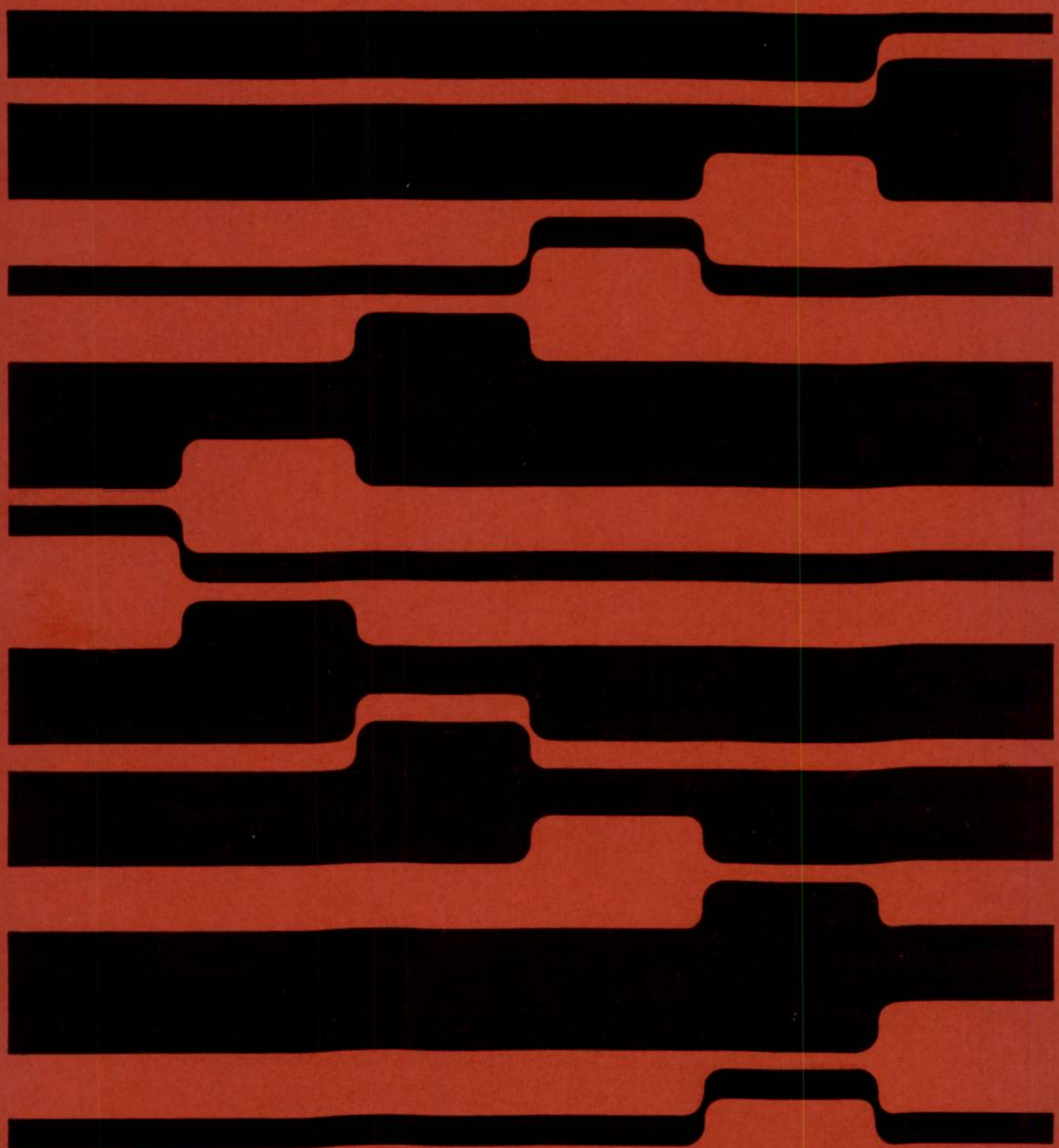
ICL

PLUTO General Description

1900 Series



ICL

PLUTO General Description

1900 Series

With effect from 9th July 1968 the name of
International Computers and Tabulators Limited
has been changed to
International Computers Limited

Technical Publication 4098

First Edition May 1968

Issued by Technical Publications Service
International Computers Limited
Head Office: ICL House, Putney, London SW15
Produced by ICL Printing Services
at Letchworth, Hertfordshire

Preface

This manual provides a general introduction to PLUTO, a powerful system for the creation, maintenance and use of direct access files which record structures. PLUTO provides a set of powerful subroutines for processing structure information, plus a suite of file reorganization programs.

The minimum configuration required to use PLUTO is:

Any 1900 series processor with 16K words of core store;

2 E.D.S. transports or 1 F.D.S.;

Console typewriter;

Basic input and output peripherals are optional, since PLUTO is not concerned with input and output. Magnetic tapes can be used, but are not essential.

This manual is intended as an introduction to the basic PLUTO concepts and methods for management, systems analysts, and programmers. Systems based on PLUTO files are described to illustrate some of the diverse applications. The way in which the user's data is held and controlled on PLUTO files, and the differences between PLUTO retrieval and conventional direct access retrieval are fully explained. The uses of the standard (high level) PLUTO retrieval routines and of the basic (low level) retrieval functions are shown by examples, and the general principles of PLUTO file reorganization are outlined.

The full details required by programmers are given in the I.C.T. manual *PLUTO Basic System*.

Acknowledgment

PLUTO was initially specified by *Telefonaktiebolaget L. M. Ericsson* and has been developed by I.C.T. in co-operation with that company.

Contents

Preface iii

Chapter 1 Introduction 1
STRUCTURES 1
RETRIEVAL 5
PLUTO FILES 9

Chapter 2 Some applications 15
A COMPREHENSIVE PRODUCTION CONTROL SYSTEM 15
CONDITIONAL STRUCTURES USED IN COSTING 17
A PURCHASE CONTROL SYSTEM 18

Chapter 3 User defined area 21
FORMAT 21
GROUP HEADERS 23
RETRIEVAL 23

Chapter 4 File maintenance 25
PLUTO INDEXES 25
FILE CREATION AND MODIFICATION 27
PLUTO LINKS 31
PRESENTATION OF DATA 40

Chapter 5 Standard retrieval 41
PARAMETERS PRESENTED TO RETRIEVAL FUNCTIONS 41
OUTPUT 41
SINGLE LEVEL RETRIEVAL 42
INDENTED RETRIEVAL 45
SUMMARIZED RETRIEVAL 49

Chapter 6 PLUTO basic functions 59
BASIC RECORD HANDLING AND RETRIEVAL 59
QUANTITY AND CONDITION FUNCTIONS 59
CHAIN FUNCTIONS 60
BASIC STRUCTURE EXPLORATION 60
THE USE OF BASIC FUNCTIONS 60

Chapter 7 File reorganization 63
USES OF REORGANIZATION 63
FACILITIES OFFERED BY PLUTO FILE REORGANIZATION 63
GENERAL PRINCIPLES 64

Index 65

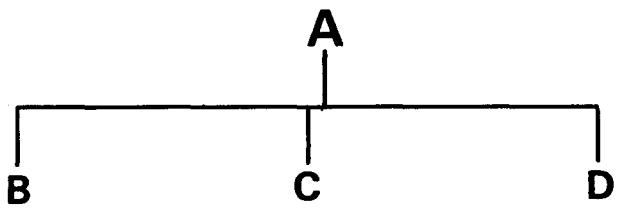


Figure 1 A tree structure

Parts structure of A.

Operations structure of A.

Figure 2 Assembly A has two structures

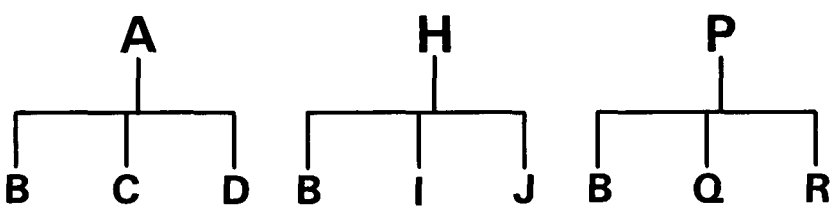


Figure 3 B is an element in several structures

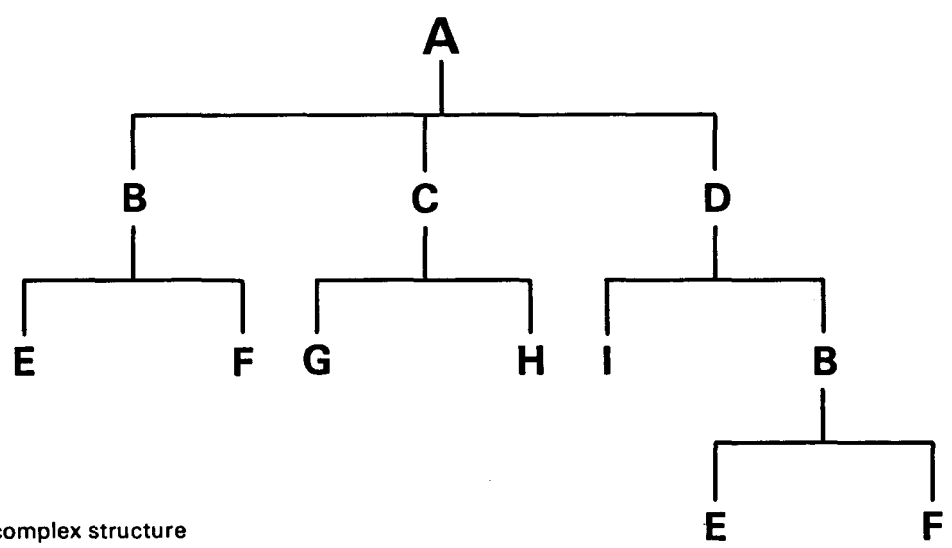


Figure 4 A complex structure

Chapter 1 Introduction

PLUTO (Parts-Listing/Used-on Technique) is a powerful system for creating, maintaining and using direct access files which record structures. This chapter introduces the concept of structures, and discusses some of the situations which they can be used to represent. The use of PLUTO files in an integrated information system is explained, with a brief summary of the different forms in which structure data can be retrieved and presented. The uses of these forms of retrieval are also outlined.

STRUCTURES

Simple structures

In Figure 1 a structure is pictured as a tree-diagram. A structure of this kind could be used to represent many things: A might be an assembly, and B, C and D its components; or A might be a part, and B, C and D the operations necessary to produce it; or A might be an article, and B, C and D alternative sources of supply for that article: these are just a few examples from the diversity of situations which can be represented as structures. In all these cases, data is represented by the links in the structure (the lines in the tree-diagram). In one case these links represent 'is part of' (thus, 'B is part of A', 'C is part of A', etc.); in another case the links represent 'is an operation for making' ('B is an operation for making A' etc.); in the third case they represent 'is a source of supply of': the meaning of the structure links depends on the user. A structure can be used to represent any situation where there is a relationship between elements.

The structure shown in Figure 1 is very simple, and can be easily visualized, and recorded on a piece of paper. More complex structures call for the more sophisticated techniques made available in PLUTO.

Complex structures

In one type of complex structure situation, a single element may have more than one structure: for example, if A is an assembly, it may have components B, C and D, and operations X, Y and Z may be necessary to make it (see Figure 2).

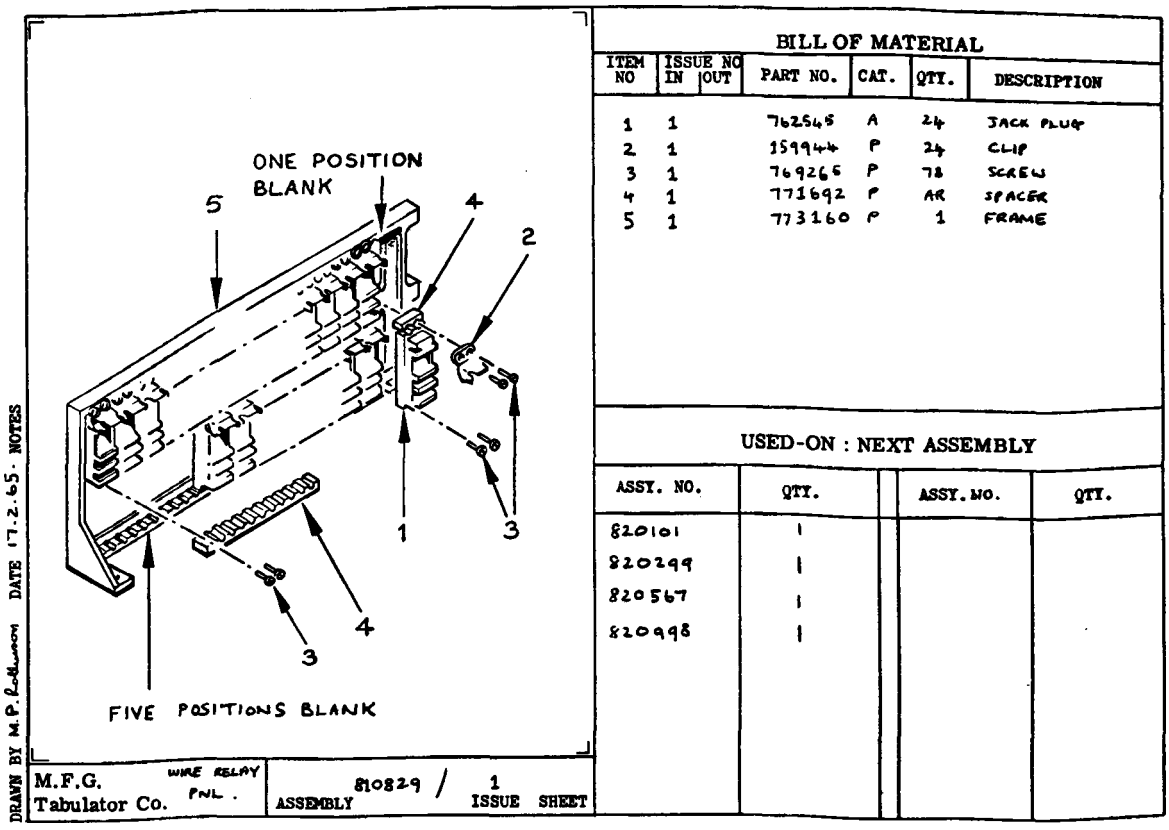
An element may also be a member of several structures of the same kind (e.g. a part may be used on several assemblies) as in Figure 3.

A structure may have several levels, and an element may occur at more than one level in the structure, as in Figure 4.

PLUTO terminology

No matter what the structure represents, certain standard terminology is used in PLUTO.

Unit	'Unit' is used as a general term for the elements of a structure. In the Bill of Materials application (page 3), the elements are parts; in the Routings application (page 16), the elements are parts and work-centres, and so on. Whatever the elements are, they are referred to as 'units'.
Component	The 'components' of a unit are all those units which are linked to that unit and are below that unit in the structure. In Figure 1 units B, C and D are the components of A.
Used-on	A unit is said to be 'used on' all units to which it is linked and which are above that unit in the structure. In Figure 1 units B, C and D are used on unit A. The units on which a unit is used are known as the 'used-ons' of that unit. In Figure 4, units A and D are the used-ons of unit B.



BILL OF MATERIAL

ITEM NO	ISSUE NO IN	ISSUE NO OUT	PART NO.	CAT.	QTY.	DESCRIPTION
1	1		762545	A	24	JACK PLUG
2	1		159944	P	24	CLIP
3	1		769265	P	78	SCREW
4	1		771692	P	AR	SPACER
5	1		773160	P	1	FRAME

USED-ON : NEXT ASSEMBLY

ASSY. NO.	QTY.	ASSY. NO.	QTY.
820101	1		
820299	1		
820567	1		
820998	1		

Figure 5 Assembly drawing with bill of materials and used-on list

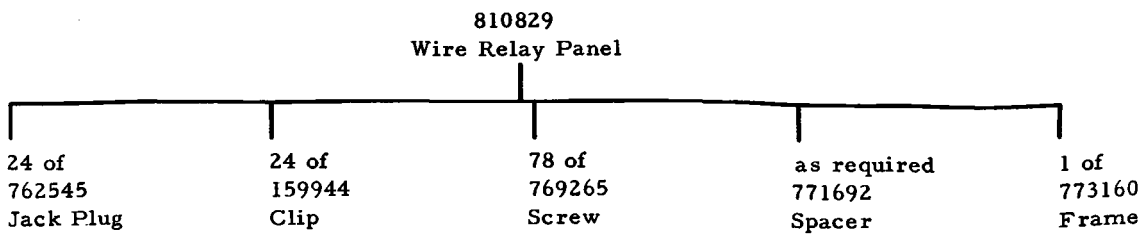


Figure 6 Component structure of wire relay panel

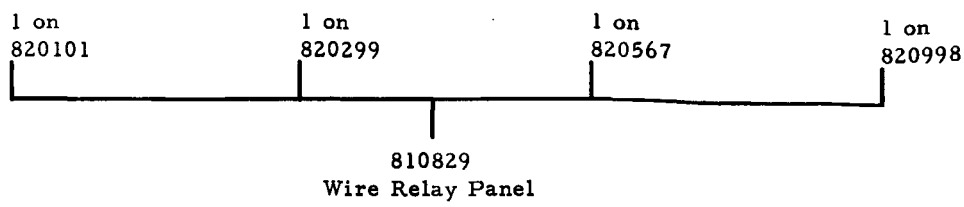


Figure 7 Used-on structure for wire relay panel

	'Where-used' and where-component' are synonyms of 'used-on'.
Direct and Indirect Components	The direct components of a unit are those components which are one level below it in the structure. The indirect components are those which are more than one level below it in the structure. In Figure 4, unit A has direct components B, C and D, and indirect components B, E, F, G, H and I. (B is both a direct and an indirect component of A).
Direct and Indirect Used-ons	The direct used-ons of a unit are those used-ons which are one level above it in the structure. The indirect used-ons are those which are more than one level above it in the structure. In Figure 4, unit G has a direct used-on, unit C, and an indirect used-on, unit A.
Parent and Subunit	Where two units are so related that one is a direct component of the other, the unit which is higher in the structure is called the parent, and the lower unit is called the subunit. Thus, in Figure 4, unit B is the parent of units E and F, and E and F are subunits of B. But A is *not* the parent of E and F, although it is a used-on of E and F; nor are E and F subunits of A, although they are components of A, because E and F are not *direct* components of A.

Integrated management information systems

In management situations where structure information is complex and affects many parts of an organization, a centralized, integrated information system becomes necessary. PLUTO enables users to design such a system using direct-access files. PLUTO uses one set of records for the units and a separate set of records for the structure links. This makes it possible to update simultaneously all structures involving a particular unit, since that unit has only one master record. Suppose Figure 3 represents the structures of three assemblies A, H and P, and B is a part used on all of these assemblies; A and H are assemblies currently being manufactured, P is at the design stage; some information on B has to be altered (e.g. B is a bought-out part, and the supplier has altered his design); if the manufacturing and design departments use separate information-systems, then two sets of records will have to be updated; but if the organization has an integrated information system, using PLUTO records, then only the unit record for B need be updated, and this immediately ensures up-to-date information to both departments.

Bill of materials application

This section illustrates the differences between PLUTO-based systems and conventional systems by reference to the Bill of Materials application.

Figure 5 shows an assembly drawing. This has been drawn on a standard form, which bears two blocks headed 'Bill of Materials' and 'Used-on: Next Assembly', in which the draughtsman enters product structure information.

The bill of materials (or parts listing) gives the direct components of the assembly. The same data could be represented by the structure shown in Figure 6.

The used-on block in Figure 5 gives the direct used-ons of the assembly. The same data could be represented by the structure shown in Figure 7.

LIMITATIONS OF CONVENTIONAL METHODS

From Figures 6 and 7 it can be seen that the data on the assembly drawing only extends one level down the structure below unit 810829 and one level up the structure above it. This will generally be as much information as it is convenient and useful to hold on the drawing itself; but clearly situations will arise which show that it is not satisfactory to use handwritten (or typed) bills of material and direct used-on lists as the basic source of product structure data. For example, in order to compute material requirements it may be necessary to access the bill of materials for each finished product, then the bill of materials for each subassembly of each product, and so on down to the bottom of the structure (in Figure 5 items 1, 2, 4 and 5 might be piece parts, and the bill of materials for each of them would then have to be accessed), and cumulative records of quantities for each unit would have to be kept; if this involves manual search and calculation, the process is time-consuming and prone to error, so that quick and reliable information cannot be obtained. Again, if a modification to a unit is being considered, it will be

necessary to consider the effects on all the products on which the assembly is ultimately used, and if this involves a manual search of the used-on lists, the process may be lengthy, and perhaps unreliable.

A partial solution to this problem is to maintain automated files, either on magnetic tape or on punched cards; however, this solution raises its own problems. For example, the records will be required both in bill of materials order and in used-on order, and so either they must be periodically sorted (in which case out-of-date information may have to be used between sorts); or duplicate files must be maintained (which doubles file-maintenance effort and introduces the risk of discrepancies between the two files). Similar problems arise with conventional direct-access systems.

PLUTO METHOD

PLUTO offers an integrated system for storing and handling product structure information. There is no duplication of records and it is as easy to explore used-ons as to explore components. In this example, the user might decide to maintain just two direct-access files, one (the master file) holding a master record for every part, subassembly and assembly (i.e. for every unit, in PLUTO terms), and the other (the structure file) recording the structure links between units. (This is a relatively simple approach: in many cases several files would be appropriate; but each unit would have only one master record.)

ADVANTAGES OF PLUTO

The adoption of the kind of system outlined above will give the following advantages:

1 Direct-access storage makes on-demand enquiries possible.

2 The maintenance of only one master record for each product or component ensures up-to-date information to the whole organization, with minimum file-maintenance effort and minimum risk of error.

3 PLUTO retrieval techniques are flexible: the outputs which the user can obtain are not unduly limited by the initial conception of the system. Additional flexibility is provided by allowing the user to write his own program for input and output, and of course to process PLUTO data in any way required before output.

Note: Where changes to a unit or structure affect the design departments before they affect the manufacturing departments, PLUTO conditional retrieval facilities enable the user to program for each to receive the appropriate data, although they use the same set of files. Confidential data can also be held on the common records, protected by security keys.

Similar advantages can be obtained in other areas of application. Chapter 2 discusses a variety of systems developed using PLUTO files. The remainder of this chapter outlines PLUTO retrieval techniques and file formats.

RETRIEVAL

Six standard (high-level) retrieval functions are provided in PLUTO, which are called single level explosion and implosion, indented explosion and implosion, and summarized explosion and implosion. Explosion functions retrieve data concerning the structure below a given unit. Implosion functions retrieve data concerning the structure above a given unit. The six functions are here explained by examples relating to the structure shown in Figure 8. Some of the possible uses of these forms of retrieval are also mentioned. PLUTO makes it easy for the user to write his own variants on these standard explosions and implosions, or to acheive any exploration of the structure (see Chapter 6, 'PLUTO Basic Functions', for further details).

Single level explosion and implosion

The single level retrieval functions enable the user to explore the immediate structure. The single level explosion of a given unit is a list of those units related to the given unit which are directly below that unit in the structure. Single level implosion is the same except that the items are directly above the initial unit.

SINGLE LEVEL EXPLOSION

The single level explosion of P1 gives the result

5 of A1	with data descriptive of A1
3 of A2	with data descriptive of A2

Single level explosion might be used, for example, to retrieve from the centralized information system the bill of materials for an assembly; the 'routing' of a part or assembly (i.e. a sequential listing of all the operations required to manufacture a part or put together an assembly) would be retrieved by a variant of single level explosion.

SINGLE LEVEL IMPLOSION

The single level implosion of B1 gives the result

4 on A2	with data descriptive of A2
1 on A4	with data descriptive of A4

Single level implosion might be used to retrieve the next assembly used-on list for a part or subassembly, or a list of all the jobs in which a particular work-centre was involved.

Indented explosion and implosion

The indented explosion of a unit is the exploration of the structure beneath the unit. The result is a list of each unit there that is directly or indirectly related to the initial unit, showing how many steps or levels come between each unit and the initial unit. In the Bill of Materials example this is a list of all components parts of the initial unit. The term 'indented' refers to the format in which such a list is commonly output.

Implosion is the same except that it is concerned with the structure above the unit instead of below the unit.

INDENTED EXPLOSION

The indented explosion retrieves the tiered structure framework of a unit. It can, for example, be used to obtain the detailed breakdown of a particular part. The user writing his own indented explosion routine can output lead times on the components.

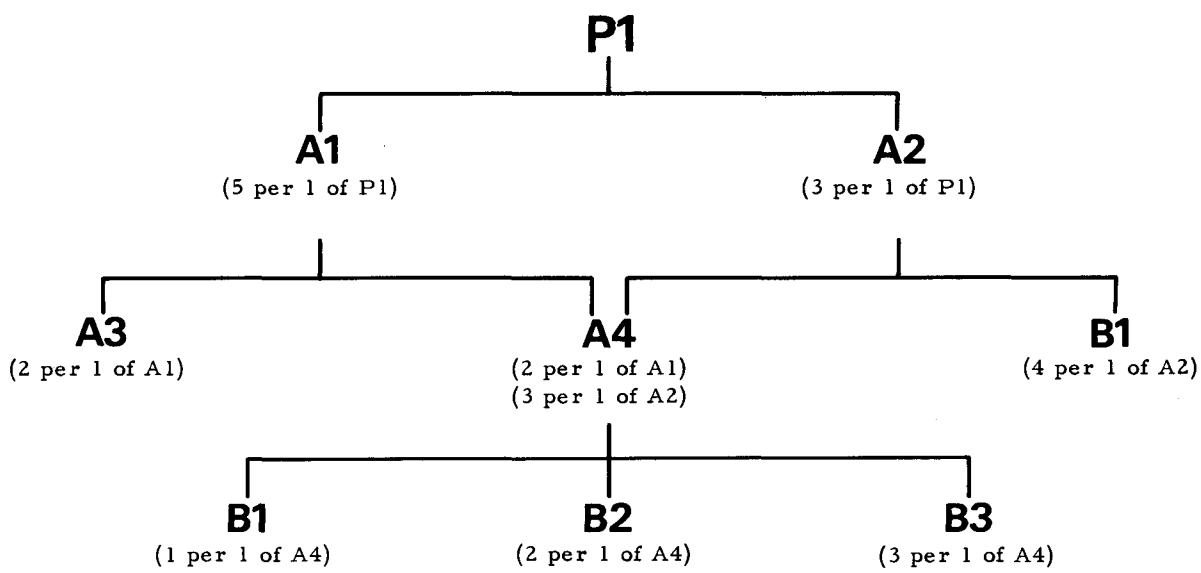


Figure 8 A structure

The indented explosion of P1 gives the result

Level	Quantity		Unit
1	5	of	A1
2	2	of	A3
2	2	of	A4
3	1	of	B1
3	2	of	B2
3	3	of	B3
1	3	of	A2
2	3	of	A4
3	1	of	B1
3	2	of	B2
3	3	of	B3
2	4	of	B1

with data descriptive of each unit.

This follows each path to the bottom of the structure in turn. Note that there are two paths to B2 and B3 so that they occur twice even though they have only one direct use (which is on A4).

INDENTED IMPLOSION

The indented implosion of B1 gives the result

Level	Quantity		Unit
1	1	on	A4
2	2	on	A1
3	5	on	P1
2	3	on	A2
3	3	on	P1
1	4	on	A2
2	3	on	P1

with data descriptive of each unit.

Indented implosion might be used in analysing the intermediate usages of a part between the next assembly where used and the end product.

Summarized explosion and implosion

The summarized retrieval functions aggregate the quantities for each unit, and create one record for each unit, with the total quantity used.

SUMMARIZED EXPLOSION

The summarized explosion of P1 gives the result

5 of A1

3 of A2

10 of A3

19 of A4

31 of B1

38 of B2

57 of B3

with data descriptive of each unit.

Summarized explosion can be used to retrieve the basic breakdown of an assembly into quantities of components; summarized explosion may be used in budgeting: the user can retrieve an estimate of annual usage of raw materials and bought out parts by presenting his annual estimate of finished goods and spares. The standard summarized explosion function does not take into account dates, stocks, etc.

SUMMARIZED IMPLOSION

The summarized implosion of B1 gives the result

2 on A1

7 on A2

1 on A4

31 on P1

with data descriptive on each unit.

Summarized implosion can be used to retrieve data on the ultimate usage of a part of assembly. For example, a list of all products which might be affected by a change in the design of a part could be obtained; summarized implosion could also be used in calculating the contribution of each part to the total cost of a product.

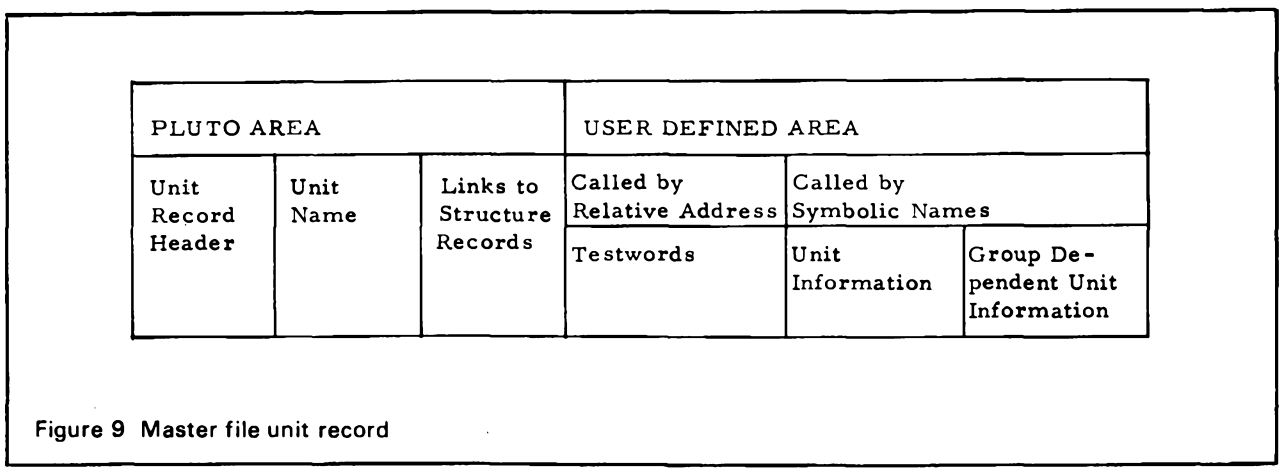


Figure 9 Master file unit record

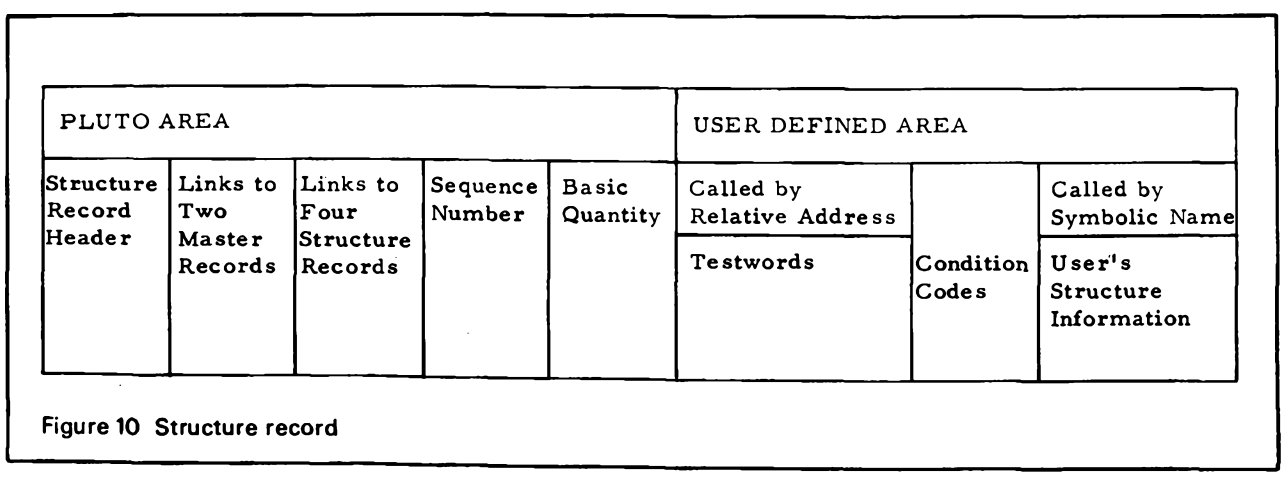


Figure 10 Structure record

PLUTO FILES

To carry information about units and about the structure, PLUTO uses two types of file: the master file and the structure file. The master file, in general, carries data about units, while the structure file, in general, carries data about the cross relationships between units. There is nothing rigid about this as PLUTO does not concern itself with the use made of the files. A master file may be referenced by a number of structure files and a structure file may reference a number of master files. PLUTO will not link two files of the same type.

A master file is indexed, enabling access to be made in the normal way. A structure file is not indexed; a structure record can only be accessed through a master file record or another structure record.

The units on a master file may be grouped – the user can then define a different record length for each group. Apart from this, PLUTO files are of fixed length records. Bucket length is user defined for each file (within the normal I.C.T. range of bucket lengths). No record may overlap from one bucket to another.

PLUTO records

Both unit (i.e. master file) records and structure records are divided into a PLUTO Area and a User Defined Area.

MASTER FILE UNIT RECORD

The format of a unit record is illustrated in Figure 9.

PLUTO area

The PLUTO area contains the following fields:

1 Unit record header: this contains data used by the PLUTO retrieval functions, which are dealt with in detail in Chapters 5 and 6.

2 Unit name: this is the unit name as defined by the user; the length of the unit name is at the user's option.

3 Links to structure records: there is a link area for each structure file with which the master file is associated. This is dealt with in greater detail below, and in Chapter 4.

User defined area

The user defined area is in two parts, the length of each being defined by the user. The first part consists of relatively addressed single words, called 'testwords'. The number of testwords is defined by the user when he creates his files. The second part is of variable length character fields called by symbolic name. These fields are known as 'unit information'. The symbolic names and lengths of the fields are defined by the user when he creates his files, and are constant throughout a file. If a file is grouped, however, some of the fields may differ from group to group. These fields are known as 'group dependent unit information'.

STRUCTURE RECORD

The format of a structure record is illustrated in Figure 10.

PLUTO area

The PLUTO area contains the following fields:

1 Structure record header: this carries information required by the retrieval functions.

2 Master record links: this field carries the direct access addresses (*not* the unit names) of the two master file unit records which are linked by this structure record.

3 Structure record links: this field carries the direct access addresses of the previous and next records on the structure chains. Each structure record can be on two structure chains: one linking all records on the file which have the same parent unit record, one linking all records on the file which have the same subunit record. The use of these chains is explained on page 11.

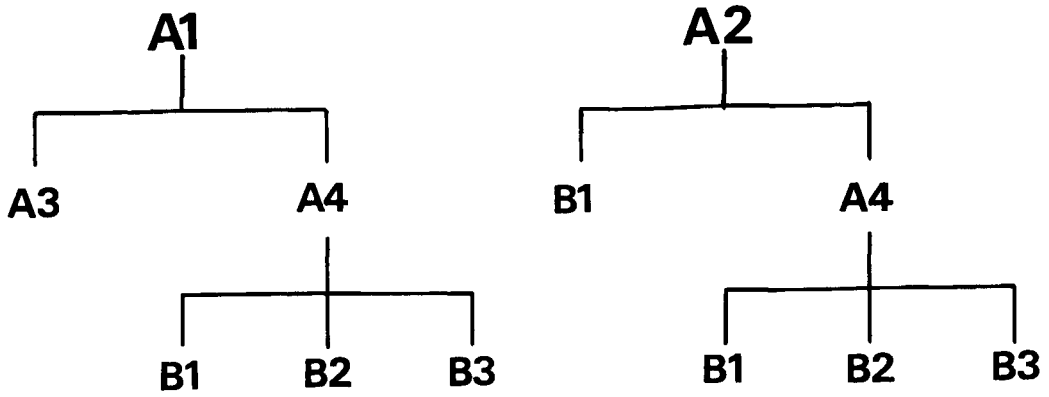
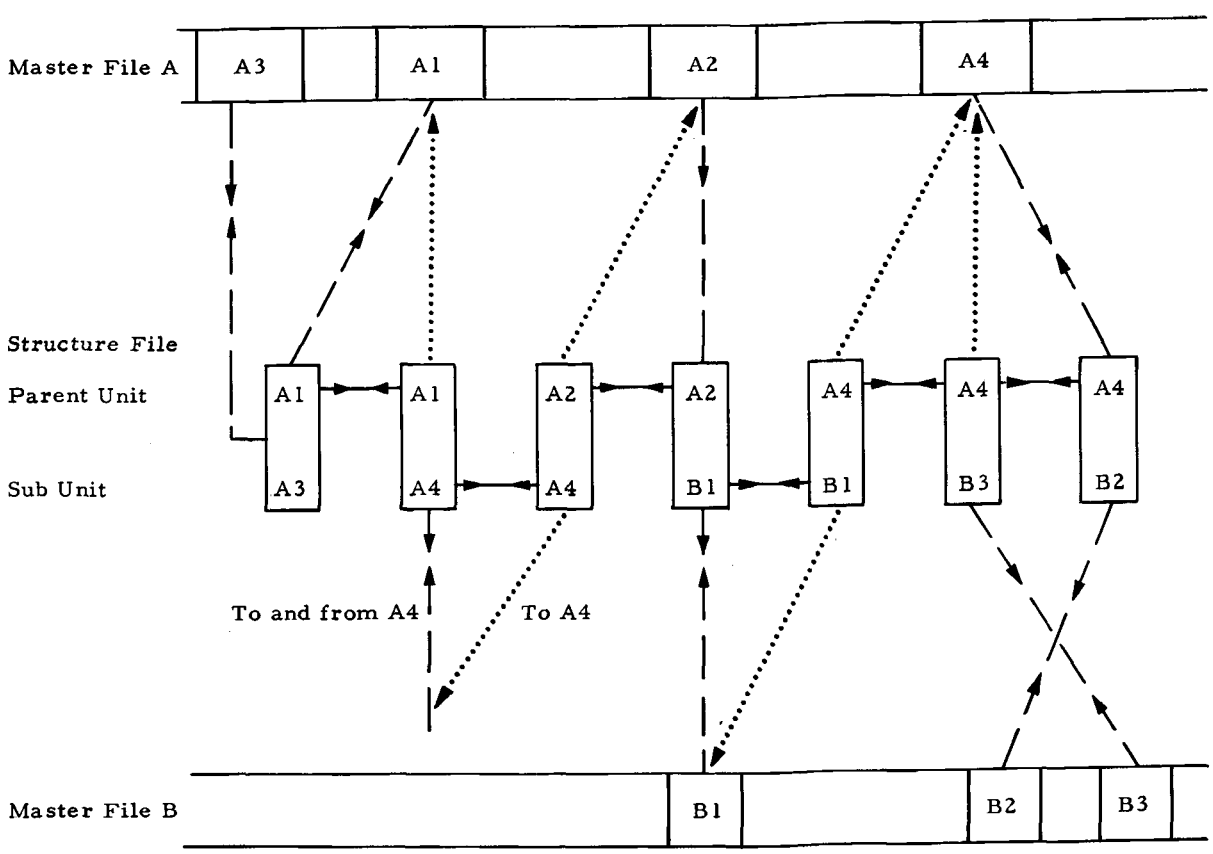


Figure 11 Two structures



Key: —— —— indicates that the master record holds the address of the structure record and the structure record holds the address of the master record.

..... indicates that the structure record holds the address of the master record.

——— indicates that each structure record holds the other's address.

Figure 12 The structures in Figure 11 represented in PLUTO files

4 The sequence number, as defined by the user: this controls the sequence of the records on the parent structure chain, and is particularly of use in the routings application.

5 Basic quantity: this is the basic quantity of the subunit required per one of the parent unit. It is held as a two-word mid-point number. In explosion or implosion it may be multiplied by a figure supplied by the user for use in conjunction with a quantity modifier held in a condition code in the user defined area.

User defined area

The user defined area of a structure record is also divided into a part consisting of relatively addressed single words and a part consisting of variable length character fields called by symbolic name. The relatively addressed single words are testwords, which may hold any data. As structure files are not indexed, the records cannot be grouped, and there is only one class of data-fields called by symbolic name; these fields are known as 'user's structure information'. Condition codes are also held in the user defined area (see Chapter 4, page 40 'Condition Codes' for further details).

PLUTO LINKS

Each master record has space for two links to each structure file referenced by the master file. One link points down the structure (towards the components), the other points upwards (towards the used-ons).

Thus access downwards from a unit's master record will obtain directly only one structure record on each of the referenced structure files. The other records which share the same parent unit are linked together (each one carries the address of its neighbours) allowing access to each one in turn. Similarly, all structure records with the same sub-unit are also linked in a chain.

'Neighbour' here does not denote physical proximity but the result of the method and sequence of insertion of the records when they are added to the file (see pages 31 and 32).

Example

The use of PLUTO links is briefly illustrated in the following example. More detailed information is given in Chapter 4.

Figure 11 shows two structures, and Figure 12 shows the PLUTO links by which they could be represented.

The direct components of A1 are A3 and A4; therefore there is a structure record to link A1 to A3 and a structure record to link A1 to A4. The unit record for A1 holds the address of only one of these structure records, in this case the record forming the link A1-A3. The structure record A1-A4 is reached by following the structure chain (shown by the heavy black line in Figure 12).

Note that for any given structure file, all the parent unit records must be on the same master file: in this example all parent units are on master file A. Subunits may be on the same master file as parent units, or on a different master file. Both cases are illustrated in Figure 12, A3 and A4 being on the parent master file, and B1, B2, and B3 being on a different master file.

Use of PLUTO files

The choice of which type of file to use for a particular function depends on the user, and upon the cross-references he requires. For example, a structure file could be used to provide effectively variable-length records, since an indefinite number of structure records can be added to the structure chain. This section describes a simple example of the application of PLUTO files to production control. Further examples are given, in more detail, in Chapter 2.

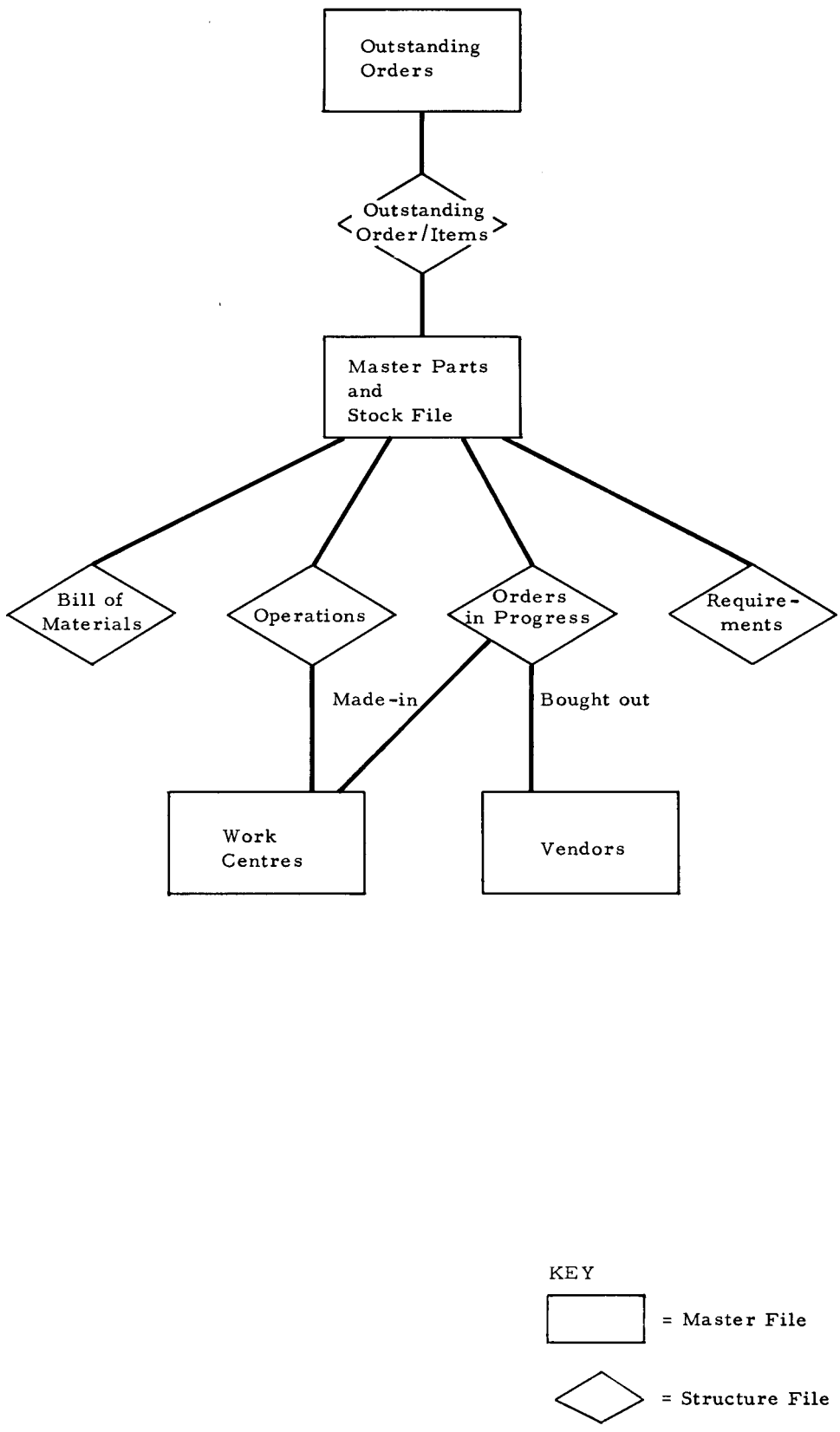


Figure 13
Example of the use of PLUTO files in production control

PRODUCTION CONTROL EXAMPLE

Figure 13 shows a simple example of the application of PLUTO files to production control. A master file of Outstanding Orders is maintained: each order is represented by a unit record; the unit name is the customer order number. Each order may require several items, and structure records on the 'Outstanding Order/Items' file will link the order master record to the appropriate items on the Master Parts and Stock File: thus a single level explosion of the order will give a list of the items required.

The Master Parts and Stock File is linked to four other structure files: Bill of Materials, Operations, Orders in Progress, and Requirements.

The Bill of Materials structure file records the product structure by linking unit records on the Master Parts and Stock File; thus a single level explosion of an assembly, using this structure file, will give a Bill of Materials for the assembly, a single level implosion will give a used-on next assembly list, an indented explosion will give the tiered structure of the assembly, a summarized explosion will give the total usage of parts and materials on the assembly, etc.

The Operations structure file has a structure record for each operation required to make each part or assembly, linking the part or assembly unit record to the unit record for the appropriate work centre on the Work Centre master file. Each Operations structure record has a part or assembly as parent: the operation structure records may be retrieved in correct sequence, according to the sequence number defined by the customer. An implosion of a work centre unit record, using this structure file, will give a list of the operations in which the work centre can be involved.

The Orders in Progress structure file links the Master Parts and Stock File to two other master files, the Work Centre file and the Vendor file, according as the order in progress involves made-in or bought-out parts. An explosion of a part or assembly unit record using this structure file would give a list of the sources of supply (internal and external). An implosion of a work centre unit record, using this structure file, would give a list of orders in which the work centre was actually involved. Note that these orders could of course arise from within the organization, (for example orders required to maintain acceptable stock-levels). An implosion of a vendor unit record would give a list of items which that vendor had currently contracted to supply: thus if, for example, a vendor's ability to supply the items was known to be affected by some contingency (e.g. fire), a list of all items affected could be quickly retrieved.

The Requirements structure file holds a series of orders not yet in progress, and a series of expected issues, and might also hold a summary of orders in progress and outstanding order items. By holding Requirements on a structure file the user virtually obtains variable-length records. In some cases the user would find fixed-length records adequate for storing Requirements, and would then hold Requirements within the unit records of the Master Parts file.

FLEXIBILITY OF PLUTO FILES

PLUTO files allow the user considerable flexibility in the design of his system.

A master file can be linked to a number of different structure files, and a structure file can be linked to a number of different master files. A structure file has one 'parent' master file: i.e. all records on that structure file hold the address of parent records on the same master file; subunits may be on any of the linked master files.

One of the user's files can be split between two PLUTO master files, both referenced by the same structure file: this would enable the user to use completely different record formats; for example records of bought-out parts might be on one file, and records of piece-parts on another. A file might also be split into two PLUTO files in order to make it possible to store a larger file on E.D.S. than would otherwise be possible, since for efficiency of processing all PLUTO files referenced by the program must be on-line. Thus the system of master and structure files in PLUTO gives maximum flexibility in the arrangement of files.

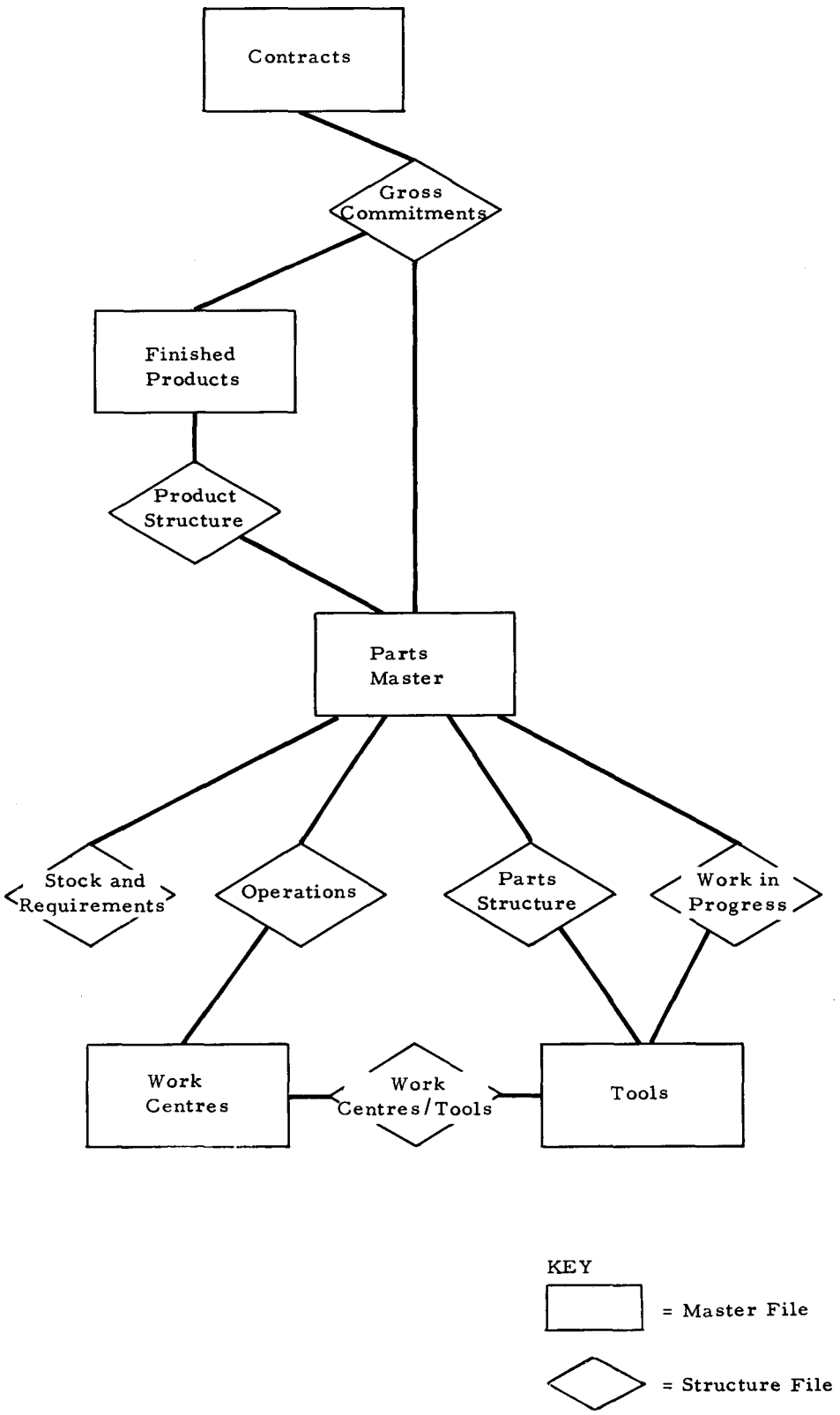


Figure 14 A production control system

Chapter 2 Some applications

This chapter describes three cases of the application of PLUTO files and retrieval techniques to different data processing requirements. As previously stated, the range of situations where structure information is involved is wide, and these examples in no way exhaust the possible applications of PLUTO files.

A COMPREHENSIVE PRODUCTION CONTROL SYSTEM

A comprehensive integrated information system for production control, using PLUTO files, is charted in Figure 14. This system uses five master files and seven structure files. The master files are: Contracts, Finished Products, Parts Master, Tools, and Work Centres. The structure files are: Gross Commitments, Product Structure, Stock and Requirements, Operations, Parts Structure, Work in Progress, and Word Centre/Tools.

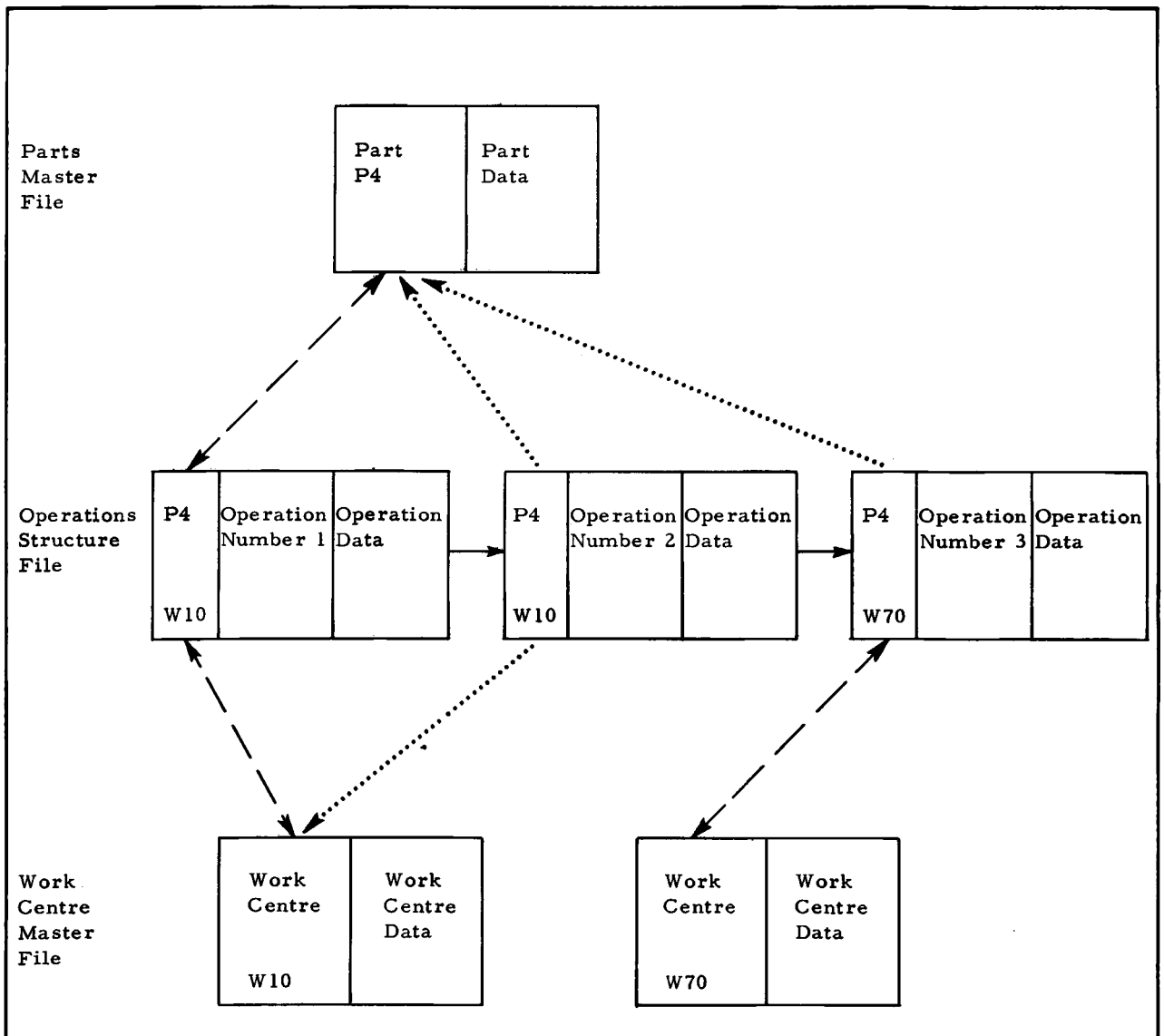
The Contracts master file carries a unit record for each contract. A contract will often involve the supply of several finished products and of spares. The Contract master record is linked to its component finished products and spares by records on the Gross Commitments structure file. In this example (as in several industries manufacturing capital equipment) the last stage of assembly takes place 'in the field', not as part of the manufacturing process, and it is therefore convenient to maintain a master file for finished products separate from the Parts Master file. What is one finished product from the customer or sales point of view is composed or several items which are finished items from the manufacturing point of view. The Product Structure file links the field-assembled product to its manufactured components on the Parts Master file. Thus the Contracts master file is linked to the Parts Master file both directly through the Gross Commitments file, which gives the parts which must be supplied as spares under the contract, and indirectly via the Gross Commitments file, the Finished Product master file, and the Product Structure file, to give the assemblies which are component items of the products to be supplied under the contract. (These items will generally be units at the highest level in the structure as defined by the Parts Structure file: see Chapter 5 for details of the method by which PLUTO handles levels in the structure.)

The Parts Master file carries a unit record for every part, subassembly, and assembly. The relations between these unit records are defined by the Parts Structure file.

Stock and requirements are defined by the Stock and Requirements structure file, on which there may be one or more records associated with each unit record on the Parts Master file. The use of a structure file for this purpose virtually gives variable length records, as explained in Chapter 1 pages 11 to 13 ('Use of PLUTO Files').

The Parts Structure file also links parts and assemblies to the tools required to manufacture them, which have unit records on the Tools master file. All records on the Parts Structure file will link a parent unit record on the Parts Master file to a subunit either on the Parts Master file or on the Tools master file. Thus through the Parts Structure file one can retrieve both the structure of a part or assembly and a list of the tools required to manufacture it. This double use of the Parts Structure file is an unusual application of a structure file in which the user holds both the parts structure and the tool links on one file using the subunit file-name (see Chapters 4 and 5) to distinguish parts structure links from tool links. PLUTO standard explosions would enable the user to retrieve the parts structure without the tools links, if required, since the program need not follow references to the Tools Master found while processing the Parts Master and Parts Structure files (see Chapter 5). If, however, the user wished to retrieve the Tool links without the parts structure, he could build his own routine using the PLUTO basic functions (see Chapter 6). The parts links of a tool can be retrieved by single level implosion.

The routing of a unit (i.e. the sequence of operations to manufacture a part or assembly) is represented by a chain of structure records on the Operations structure file. As noted in Chapter 1, (pages 9 to 11 'Structure Record'), the user defines the order of the records on the parent structure chain by means of



KEY — — — indicates that the master record holds the address of the structure record and the structure record holds the address of the master record.

····· indicates that the structure record holds the address of the master record.

———— indicates a link in the parent structure chain (i.e. the structure records are linked in the order indicated by the arrows).

Note: The Operation Number is the PLUTO sequence number.

Figure 15 Routing

the sequence number with which the structure record is presented to the PLUTO structure record creation function. The Operations structure file should always have the Parts Master file as its parent master file (see page 11 'PLUTO Links'), since it is the sequence on the parent structure chain which is controlled by the sequence number. The operations structure records also link the parts unit records to the work centres where the operations are performed. In Figure 15, for example, the operations required to manufacture part P4 are represented by the structure records on the chain. The address of the first record on the chain (i.e. the first operation) is held in the unit record for P4. The remaining records on the chain may be retrieved in order by following the chain. Each structure record links P4 to a work centre and also carries data relating to an operation. If there is more than one operation on P4 involving a given work centre, then that work centre will be linked to P4 more than once, as is the case with W10 in Figure 15.

The Tools master file is linked to the Parts Master both through the Parts Structure file and through the Work in Progress structure file. The Parts Structure file carries relatively permanent data about the possible usages of tools on the manufacture of parts: alternatives could be represented by the use of condition codes in the structure records. The Work in Progress file is a relatively volatile file, showing the actual usages of tools.

The 'Work Centres/Tools' structure file links the tools to the appropriate work centres.

CONDITIONAL STRUCTURES USED IN COSTING

An electrical manufacturer in the world market must take into account many conditions affecting the design of his product, climate and power supply being two of the more obvious. When tendering for an order he must quickly cost for such variants and, where possible, use past performance as a guide.

This can be handled by the use of PLUTO files and a questionnaire approach. One parts master file, which also holds material and labour costs, is associated with a structure file, recording the product structure. Many of the structure links are conditional.

The salesman completes a questionnaire, which has items such as those shown in Figure 16. The product number and the data from the completed questionnaire are input to a program which calls the standard PLUTO explosions. Where an assembly has different components for different climates then a condition will appear in the structure records. The standard explosions will refer to the questionnaire data and select or reject a component accordingly. Where voltage is important, for example, the structure record can say 'use if voltage is greater than 200', and so on. Some components may be selected only for a combination of conditions, for example 'temperate, AC, and 220-240 volts'. This is catered for by putting several conditions on the structure record, to form one compound condition which must be satisfied for the components to be used. Some components may always be used, but in different quantities for different conditions. This requirement too can be included in the structure record.

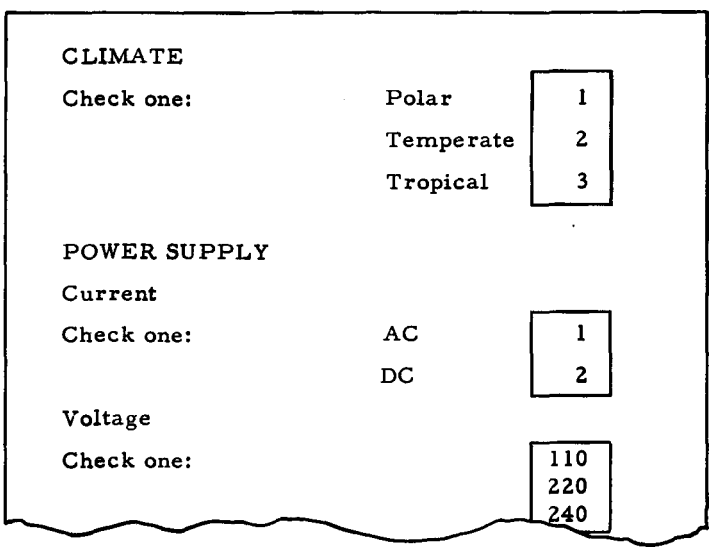


Figure 16 Questionnaire for use with conditional structures

A PURCHASE CONTROL SYSTEM

Figure 17 charts a PLUTO-based purchasing control system, using two master files linked by two structure files. One master file holds a unit record for each Item, the other holds a unit record for each Supplier. The two structure files are 'Potential Supplier Links' and 'Orders Placed'. The Potential Supplier structure records link parent unit records on the Items master file to subunit records on the Suppliers master file (i.e., show which suppliers can supply which items) and also carry data on discount per part offered by that supplier, and Vendor Rating for that item and supplier. The Orders Placed structure file records all outstanding orders by linking together the appropriate Item and Supplier for each order, also holding quantity, terms, delivery dates, etc., for each order.

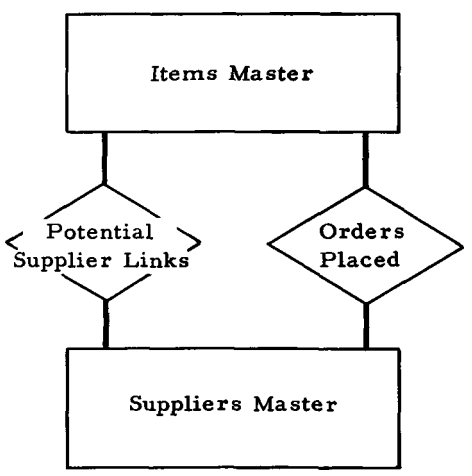
Vendor rating

The vendor rating will be based on a number of criteria, such as past performance (delivery, reliability of item, service), relation to own company, reciprocal dealings with own company, etc. Thus the vendor rating is not a function of the supplier alone nor of the item alone, and it is a considerable advantage to be able to carry the vendor rating on the structure record which links the item to the supplier.

Bulk purchasing

Taking advantage of bulk discounts offered by suppliers, or minimizing transport costs, involves consolidation of orders. A PLUTO-based purchasing control system enables the user to generate lists of suppliers with which he can match his requirements in the most economical manner (see Figure 18). The user inputs a list of items required. A PLUTO single level explosion of these items, using the Items Master file, the Potential Supplier Links structure file, and the Suppliers Master file, gives a list of sources with bulk discount information and transport costs. Next, a PLUTO single level implosion of these supplier unit records using the same three files gives a list of all the items which these suppliers can supply. Thus for each supplier who can supply any of the items required, the user has a list of all the items that supplier can supply, and bulk discount and transport information, (the Sources/ Items file).

The user's program will then match the sources to the requirements in such a way as to take most advantage of discounts and transport economies, print out the orders, and call PLUTO to write records for the orders to the Orders Placed structure file.



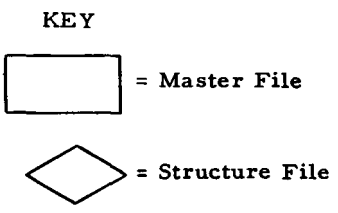


Figure 17
A purchase control system: a relatively permanent structure file links the items to the vendors who can supply them; a more volatile structure file records the orders placed by linking the items ordered to the supplier. See text and Figure 18

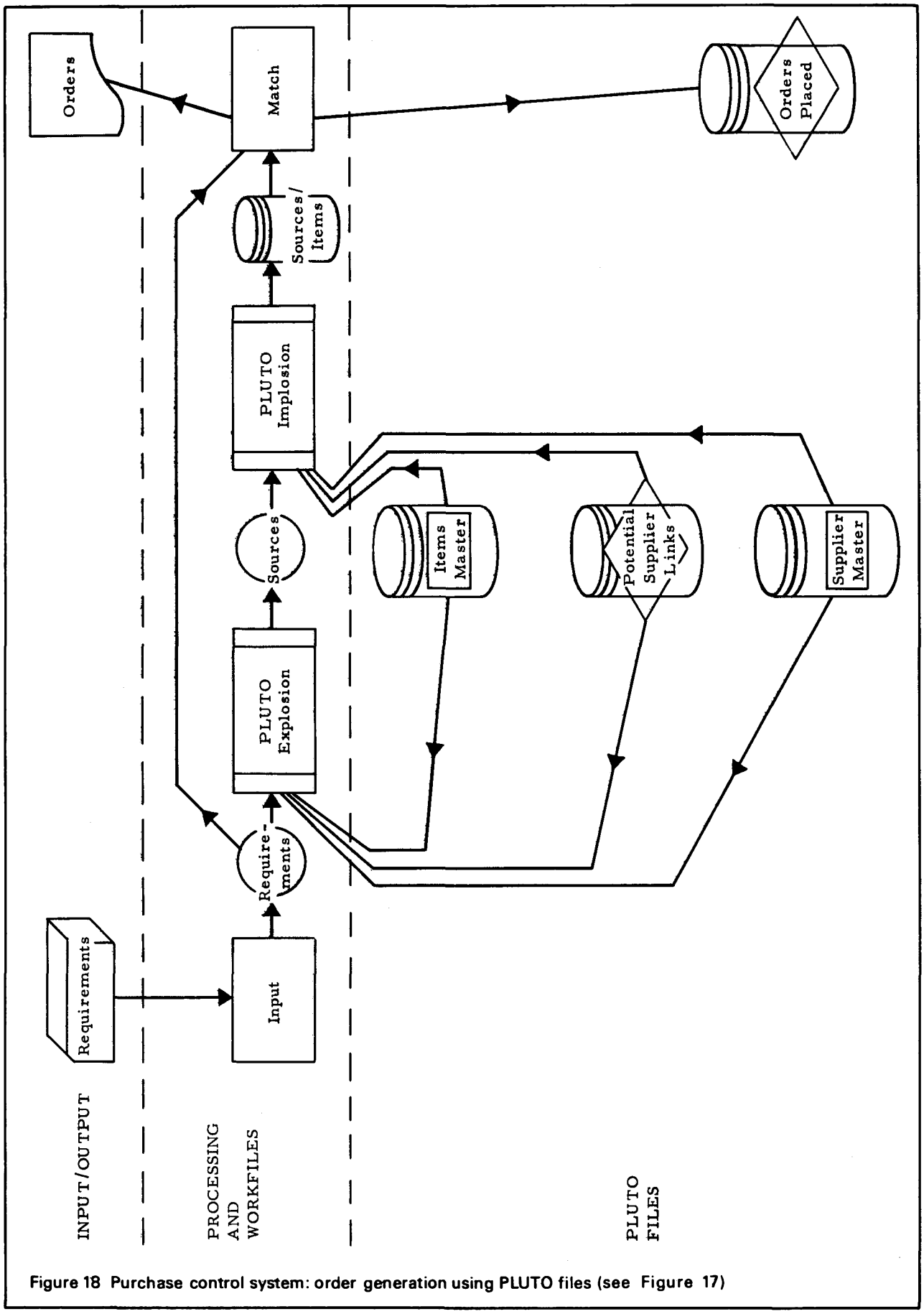


Figure 18 Purchase control system: order generation using PLUTO files (see Figure 17)

MASTER UNIT RECORD

PLUTO AREA	USER-DEFINED AREA		
	Called by Relative Address	Called by Symbolic Names	
	Testwords	Unit Information Fields	Group Dependent Unit Information Fields

STRUCTURE RECORD

PLUTO AREA	USER-DEFINED AREA		
	Called by Relative Address		Called by Symbolic Names
	Testwords	Condition Codes	User's Structure Information

Figure 19 Record formats

Chapter 3 User defined area

This chapter explains how the user's data is held and controlled on PLUTO files.

FORMAT

Both master and structure file records are divided into a PLUTO area, which contains the information which defines the relationship to other records and holds other information required by PLUTO, and a User Defined Area, which can carry whatever data the user requires (see Figure 19).

When creating PLUTO files the user specifies the record format for the user defined area. This specification is written to the file and is used by PLUTO programs to handle the records.

The formats can be altered considerably by PLUTO File Reorganization routines, which preserve as much of the data on the file as is meaningful in the new format (see Chapter 7).

The data fields in the user defined area on both unit records and structure records fall into two classes:

1 one-word fields called by relative address, which are known as 'testwords',

2 variable-length character or word fields which are called by symbolic names.

Testwords

Testwords are data fields of one word length, called by their relative address. The number of testwords on each record is defined in the file header by the user when creating his files.

Testwords are so called because data to be used for comparison conditions must be held in these fields: but it is important to realise that testwords may be used to hold any data whatsoever, according to the user's requirements. They will generally be the most convenient way of holding numeric data. On a structure record, any modifier by which the basic quantity is to be multiplied must be held in a testword.

The contents of testwords are protected by security keys, which must be presented to the PLUTO write or read function together with the relative address of the testword.

Note: Condition codes in structure records are also held in the user defined area. They have a special significance, as explained in Chapter 4 (page 40).

Named fields

The format of those data fields which are called by symbolic names is defined by strings of characters, in file and group headers, known as layout strings. When creating files, the user defines the number of characters or number of words in each field, and assigns a symbolic name. Repeatable fields may also be specified: on file creation, the maximum number of repetitions is stated on presentation of the data to PLUTO. On retrieval, the user need only specify the symbolic names of the fields which he requires. PLUTO will ensure that the requisite layout string is in core and will decode it automatically. The use of these fields is again the user's option, but clearly the main use for character fields is descriptive matter.

A single PLUTO field may contain several fields of the user's data, which the user isolates in the conventional way after retrieving the PLUTO field.

MASTER FILE

On a master file record, the data fields addressed by symbolic names are known as 'unit information'.

In general, the format of unit information will be identical throughout a file, but if records have been formed into groups, certain fields may only have identical format throughout the group: these fields are known as 'group dependent unit information'.

STRUCTURE FILE

A structure file is not indexed: therefore records are never grouped, and there is only one class of data called by symbolic names, which is known as 'user's structure information'.

Use of PLUTO data fields

Testwords will generally be used to hold numeric data, and named fields to hold non-numeric data.

MASTER FILE

A simple stock-file application of PLUTO files will serve to illustrate the different uses of testwords and unit information fields. Testwords would be used to carry such data as;

Standard batch size

Scrap allowance

Unit cost

Cycle time

Physical stock

Safety stock.

Unit information fields would be used for such data as:

Stock group code

Responsibility code

Issue denomination

Receipt denomination

Store location code.

Group dependent information fields would be used for data which varied from group to group within the file. For example, unit records on a master file might be grouped into bought-out parts, piece parts, and assemblies; certain data fields would be required only for records within a particular group, as shown below:

Group	Group dependent information
Bought-out parts	Vendor data
Piece parts	Machine shop data
Assemblies	Assembly shop data

STRUCTURE FILE

Structure records may be used to carry information, as well as linking unit records together. For example, the sequence of operations necessary to manufacture a unit could be recorded on a series of structure records associated with the unit's master record.

Each structure record could carry the following information for one operation:

Data	Area
Sequence number	PLUTO Area
Process time (one-off)	Testwords
Set up time	
Cost rate	
Interval time	
Operation description	User's structure information
Work centre description	

GROUP HEADERS

The group header carries information which relates to the group as such. This includes the layout string for group dependent information. Data which is standard for all the records in a group can also be held in the group header, in order to save space.

RETRIEVAL

Low level retrieval

PLUTO basic retrieval functions enable the user to move into a core area a single testword, or field of unit information, group dependent information, or user's structure information, from a single record. Using these functions, the user can build up his own retrieval program. Further details are given in Chapter 6.

High level retrieval

The user may present to the standard retrieval routines a list containing the numbers of testwords and names of named fields from which the contents are required. PLUTO outputs to a workfile the explosion or implosion required, with the contents of the data fields requested, for each unit. Further details are given in Chapter 5.

Chapter 4 File maintenance

The PLUTO file maintenance functions enable the user to create and modify his files. PLUTO maintains indexes, low level codes, and links between records. The user marshals the data in core store and calls the appropriate PLUTO functions, with a parameter which includes the address of the data.

PLUTO INDEXES

Only master files are indexed. A structure record can only be accessed through an associated master unit record which bears its address, or by following a structure chain from a structure record so accessed.

For reasons of economy, since unit names are often lengthy, PLUTO structure records do not use unit names to access master records: a master record is identified by record number within bucket number. This system is not compatible with bucket indexing, since if records were stored in key (i.e. unit name) sequence then record numbers within the bucket would often change when a new record was inserted, and extensive changes to links would become necessary. Consider, for example, a bucket containing all records whose unit names fall in the range 11 to 20. Records for units 11, 13 and 17 already exist. All structure records referencing unit 17 will hold the bucket number, and record number 3 within the bucket. If a record is now created for unit 16, the record for unit 17 becomes record number 4 within the bucket, and all structure records referencing unit 17 will have to be altered. This could involve processing up to 15 different files. To avoid extensive link changes on insertion of new records, PLUTO does not file unit records in unit name sequence. PLUTO indexes therefore give record numbers within bucket number for each unit record.

Grouping of records

Unit records on a master file may be combined into groups, according to the leading characters of the unit names. These characters form the group name. The user defines groups and group names when setting up his files.

A grouped file has a supervisory index, which is an index of group headers. For each group there is a group index, which gives the bucket number and record number of each unit record.

An ungrouped file has no supervisory index, and only one group index.

Two-level indexes

Both the supervisory index and the group indexes have two levels.

In the supervisory index, groups are combined into sets of groups, whose group names fall within a certain range, and the first level is an index of sets of groups: for each set of groups the first level holds the address of one bucket at the second level. The second level is an index of group headers: for each group, the bucket number of the group header is entered in the second level of the supervisory index.

In the group index, units are confined into sets of units, whose unit names fall within a certain range. The first level of the group index is an index of sets of units: for each set of units, the first level of the group index holds the address of one bucket at the second level. The second level holds the bucket and record number for each unit record.

Summary:	Supervisory Index Level 1:	entries for sets of groups.
	Supervisory Index Level 2:	entries for groups.
	Group Index Level 1:	entries for sets of units.
	Group Index Level 2:	entries for units.

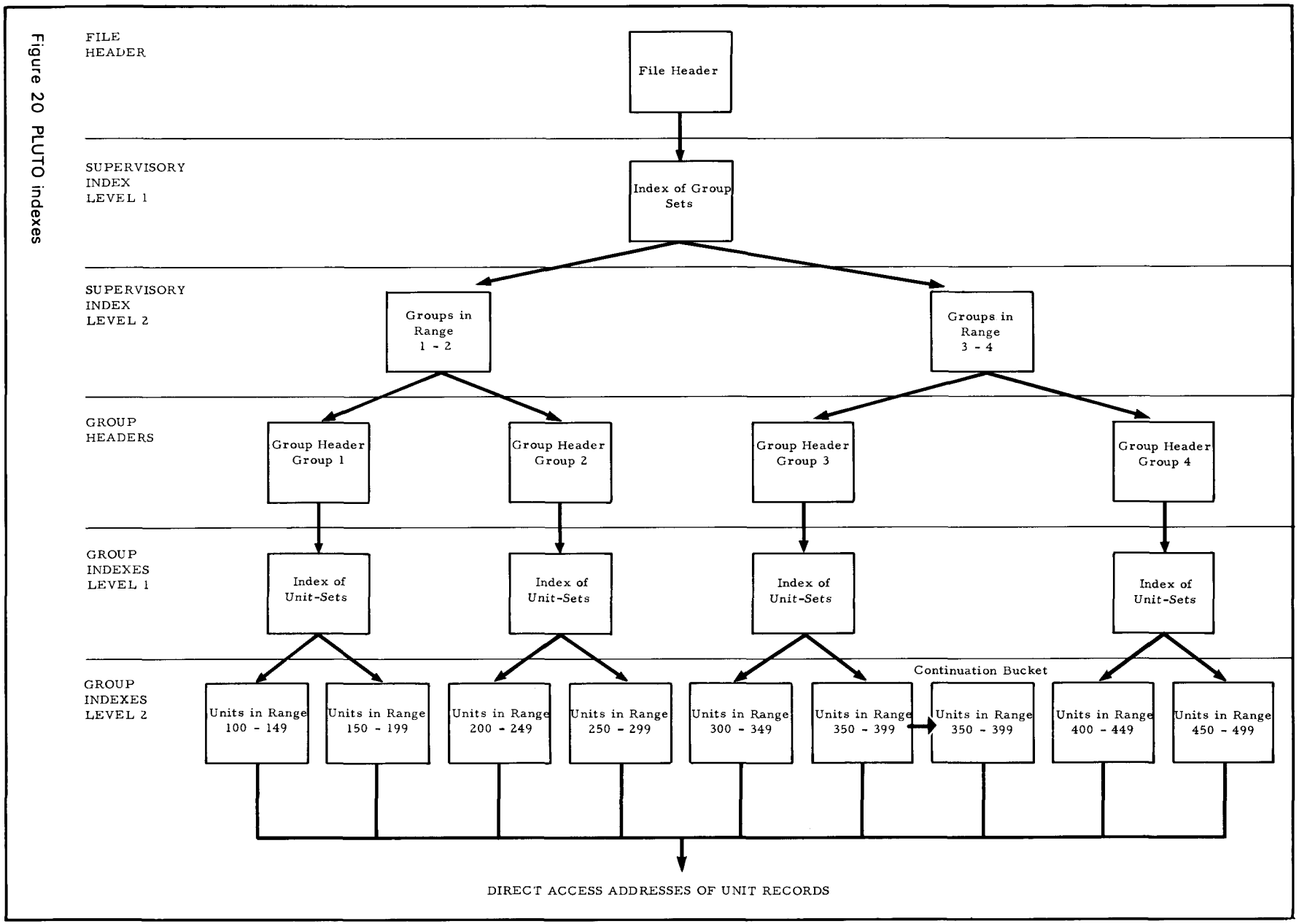


Figure 20 PLUTO indexes

If a file is grouped, the file header holds the address of the first level of the supervisory index, and the group header holds the address of the first level of the group index. If a file is not grouped, the file header holds the address of the first level of the group index.

Thus, as shown in Figure 20, the order in which the index is searched on a grouped file is as follows: File header, Supervisory Index Level 1, Supervisory Index Level 2, Group Header, Group Index Level 1, Group Index Level 2, Unit Record. On an ungrouped file it is: File Header, Group Index Level 1, Group Index Level 2, Unit Record. The number of disc accesses required depends on the user's choice of buffers and on the sequence in which units are processed.

Continuation buckets

Maximum efficiency is achieved when sets of groups and sets of units are so defined that the first level of each index occupies only one bucket and the second level occupies one bucket for each set. However, PLUTO will open continuation buckets where necessary. Figure 20 shows a continuation bucket in the second level of the group index.

FILE CREATION AND MODIFICATION

File creation

The user defines his record formats for the files by means of the two file creation functions, PLUTO Create Master File and PLUTO Create Structure File. These functions read data to the File Header, set up index areas and check file allocation and establish file connections.

HEADER DATA

The file header carries the basic housekeeping information for the file. This includes the number of testwords, the layout strings which define the character fields in the user defined area, and all other data required to define the record format. Details are given in the I.C.T. manual 'PLUTO Basic System'.

The user sets up the header information in an area in core store, and calls the appropriate function with a parameter which includes the address of the header information. PLUTO validates the header information against the File Allocator parameters (see *1900 Library Specifications*), and sets up the header.

INDEX

The master file creation routine also reserves file areas for the supervisory index.

FILE CONNECTION

Before a structure file is created, the master file to which it is linked must be created. Before a structure record is written the units it links must already have been written to the files.

Loading PLUTO files

Data is added to PLUTO files by two functions 'PLUTO Write to Master File' and 'PLUTO Write to Structure File'.

MASTER FILES

'PLUTO Write to Master File' writes group data or unit data to a master file, and makes appropriate entries in the supervisory and group indexes.

Group data

The information given is validated and expanded to form a group header. The address of the group header is then entered in the supervisory index, the group header is written to the file, and the file area for the group index is reserved.

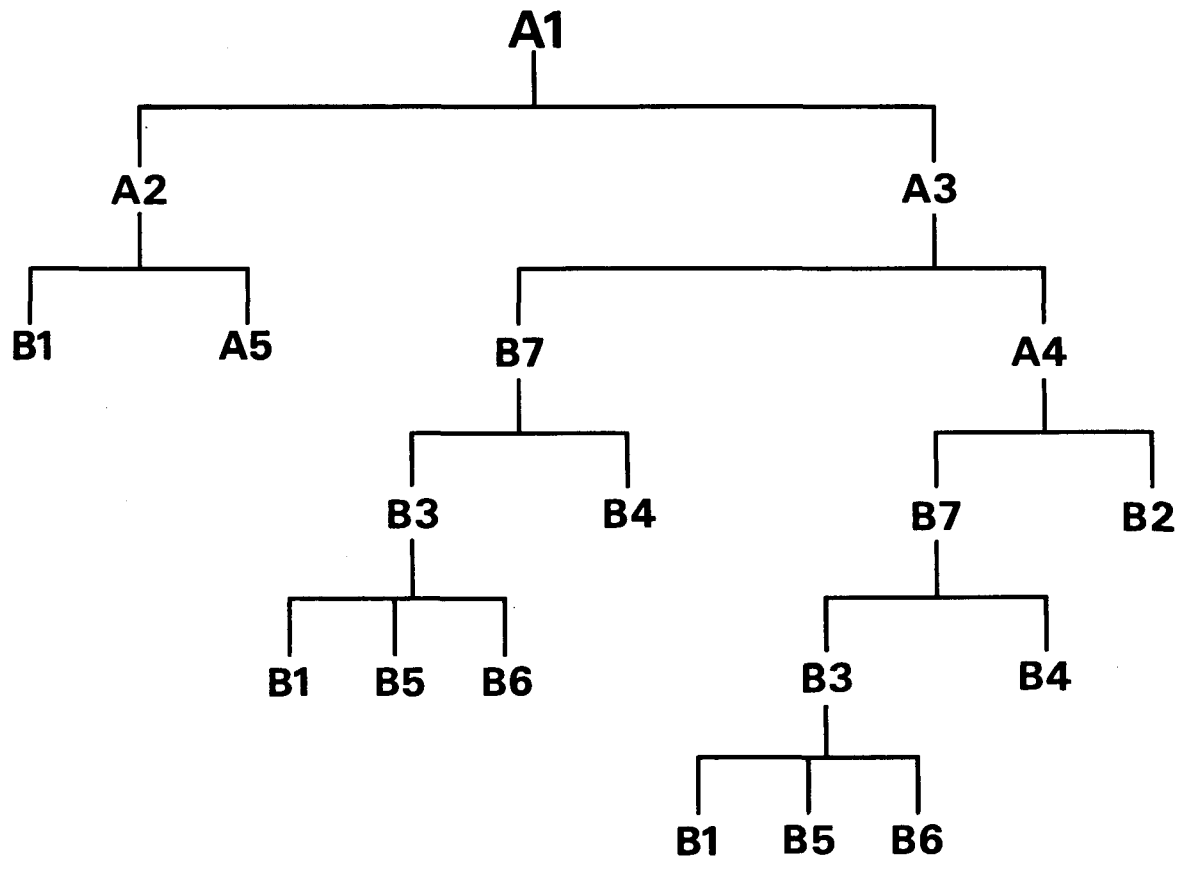


Figure 21 Complete structure

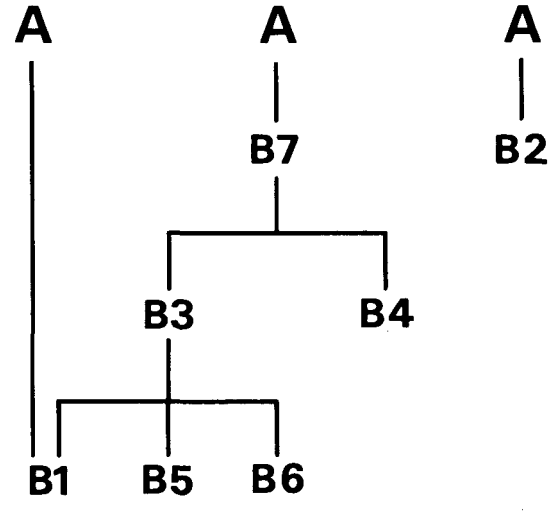


Figure 22 Structure within master file B

Unit data

Unit data cannot be written to the file unless the group index is already in existence.

The file is opened and the group index is obtained by searching the supervisory index. If the unit name is already in the index, the data is rejected and an error report is given. Using the layout strings, test-word security keys etc. in the file header, the data presented is expanded into a unit record. This record is then added to the file at the next free address. The new unit record is entered in the group index.

STRUCTURE FILES

'PLUTO Write to Structure File' writes individual structure records to a structure file, and sets links to master file unit records and links from unit records to structure records. In addition, links in the structure chains are set, low level codes are calculated and set in the unit records and low level chains generated in the parent master file (see below 'Low level codes and chains'). Before this routine can be called, the structure file itself must have been created, and all unit records to be linked must have been written to master files.

Low level codes and chains

The low level code of a unit record indicates the lowest level at which the unit is found in any structure within the master file. That part of a structure whose units are on a different master file is ignored in calculating the low level code. For example, the structure shown in Figure 21 has units on two master files, 'A' and 'B'. In calculating low level codes for master file B, level 1 is taken to be composed of all units which do not have a parent on master file B. Thus the calculation deals with a simplified structure as shown in Figure 22; level 1 consists of B1, B2 and B7, which all have a parent on master file A; and the fact that B7 is used on both A3 and A4, which are on different levels, is of no importance for calculating low level codes since A3 and A4 are on a different master file. The low level codes on B are as follows:

Unit	Low level code
B1	3
B2	1
B3	2
B4	2
B5	3
B6	3
B7	1

A unit which had no parent would be on level 1, and have a low level code of 1. If several structure files reference a master file, a unit may be an element in several structures. Its low level code is then the lowest level it occupies in any of these structures.

The use of low level codes in summarized explosion is explained in Chapter 5.

Low level chains link all unit records on the master file which have the same low level code.

Changing data on PLUTO files

Data on PLUTO files may be altered by two functions 'PLUTO change Information on Master File' and 'PLUTO change Information on Structure File'. These functions may not be used to alter record formats. The number of repeats of fields used may be altered. Record formats can only be changed by the File Reorganization program.

MASTER FILES

'PLUTO change Information on Master File' can change the contents of a group header or of the user-defined area of a unit record or records. In the latter case there are three options: the change may be made to a specific unit record, to all unit records in a group, or to all unit records on a file. Separate calls of this routine must be made to insert a new value in each field or testword. Naturally the record is kept in core until all changes have been made.

MASTER FILE UNIT RECORD LINK AREA

There is an area for each structure file associated with this master file in the order in which the structure file names appear in the master file header.

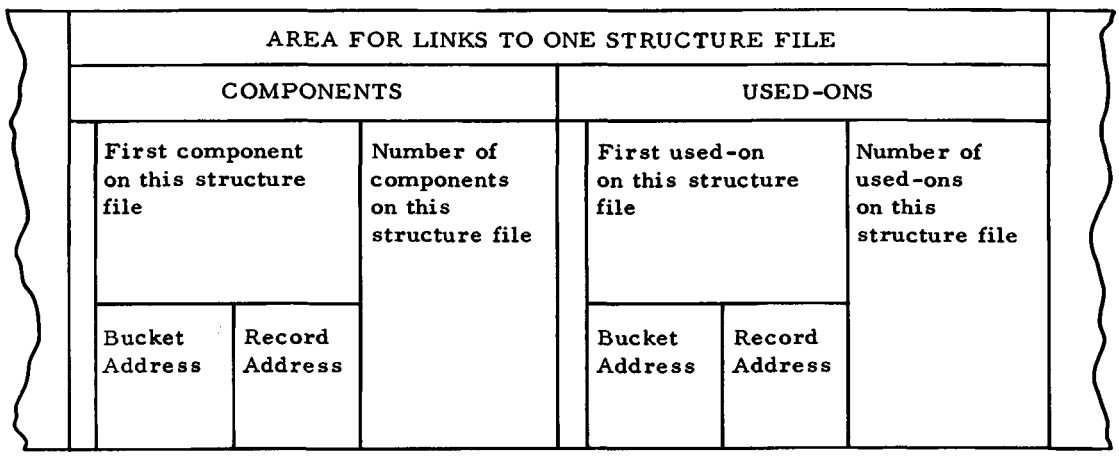
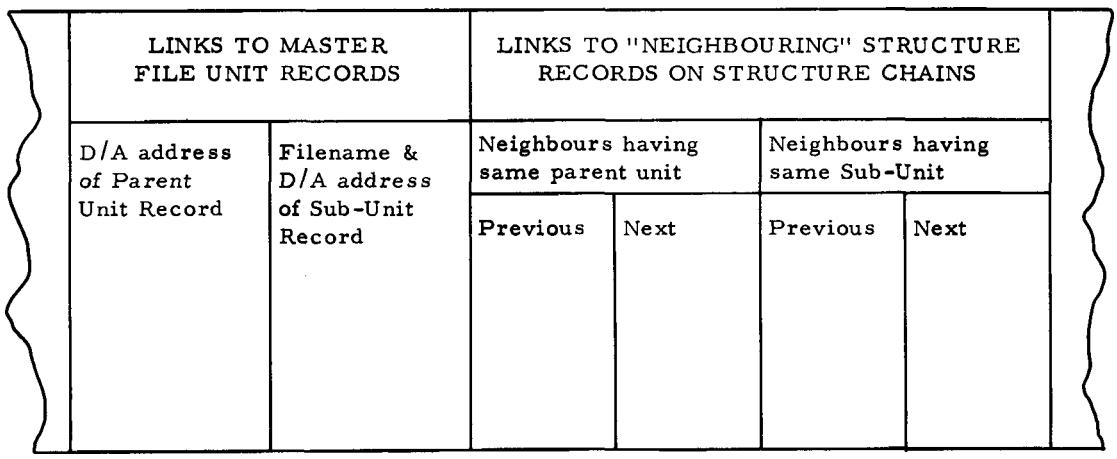


Figure 23 Master file unit record link area

STRUCTURE RECORD LINK AREAS

There are two link areas on a structure record: one for links to unit records, one for links to structure records.



NOTE: This represents the logical relationship among the links, not their physical order on a structure record. (See the I.C.T. manual"PLUTO Basic System" for formats).

Figure 24 Structure record link areas

STRUCTURE FILES

'PLUTO change Information on Structure File' can alter information on a specific structure record, on all structure records on a parent chain, or on all structure records on a subunit chain. This routine can change data affecting condition codes and quantity modifiers, sequence of structure records, or user defined information. A change in sequence of structure records will involve changing the links from structure record and may, in addition, involve a change in the link from the parent unit master record. These changes are made automatically.

Erasing records from PLUTO files

Two functions, 'PLUTO Erase records from Master File' and 'PLUTO Erase records from Structure File', erase records by setting markers in them. These records remain in the file until deleted by the File Reorganization program.

MASTER FILE

'PLUTO Erase records from Master File' can erase a single unit record; or a whole group by setting an erase marker in the group header as well as in all unit records in the group.

STRUCTURE FILE

'PLUTO Erase records from Structure File' can erase a single structure record, or all structure records with the same parent unit or all structure records with the same subunit.

PLUTO LINKS

This section describes the links maintained by the file maintenance routines and explains their function.

Master file links to structure files

The master file header carries a list of all structure files to which unit records on the master file can be linked. Each unit record has a four-word link area for each of the structure files listed in the master file header. The order of the link areas in the unit record corresponds to the order in which the structure files are listed in the master file header.

Figure 23 shows the format of the area carrying the links from one unit record to one structure file. The link area holds addresses of only two structure records, one pointing down the structure (towards the components) and one pointing up the structure (towards the used-ons). The other components or used-ons are traced by following links within the structure file. The link area on the unit record also records the total number of structure records associated with the unit record at each of the adjoining levels.

Structure record links

The structure file header carries a list of master files with which the structure file is associated.

Figure 24 shows the link areas on a structure record.

LINKS TO MASTER RECORDS

A structure record carries the address of two master records: the parent unit record (the higher in the structure) and the subunit record. These are the records of the two units whose relationship in the structure is defined by this structure record. For a given structure file all parent unit master records must be on the same master file, but a subunit may be on any of the referenced master files.

LINKS TO NEIGHBOURS

A structure record also carries links to four other structure records: the previous and the next record having the same parent unit record, and the previous and the next record having the same subunit record. These records are known as this record's neighbours.

All the structure records with the same parent are linked, either directly or through their neighbours. The series of links which link each record to its neighbours form a chain of records, known as the 'structure chain'. Each record will be on two structure chains, the parent structure chain which links all structure records having the same parent, and the subunit structure chain which links all structure chains having the same subunit. The parent unit record is linked to one end of the parent structure chain; by following the next link in each record all records can be accessed in turn. Similarly, the subunit master record is linked to one end of the subunit structure chain. The sequence of accessing is therefore not dependent on the physical sequence of the structure records on the disc: it is controlled by how the links are set. Inevitably either the parent structure chains or the subunit structure chains must clash with physical sequence. After PLUTO file reorganization the parent chains will coincide with the physical sequence, but this will no longer be the case when more records have been added.

The structure chains are followed by the PLUTO explosion and implosion routines. The explosion routines follow the parent structure chain, beginning at the first component structure records (i.e. the first structure record linking the parent to a component) whose address is given in the parent unit record. The implosion routines follow the subunit structure chain, beginning at the first used-on structure record, whose address is given in the subunit master record. Links are set up by the PLUTO write functions as the records are added to the file. The logical sequence on a chain of structure records with same parent is controlled by the sequence number which the user supplies with the data to the write function ('PLUTO Write to Structure File').

Purpose of PLUTO links

Figure 25 shows the structure of a unit A4, which is a subassembly in the two structures illustrated in Figure 11 (Chapter 1, Page 10).

Figure 26 shows how this structure could be represented on PLUTO files. Figures 27 to 31 with their associated text show how the structure can be retrieved by following the links (indicated by arrows in the figures).

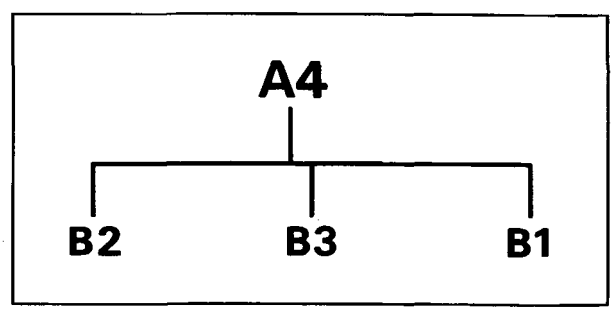


Figure 25 Structure of unit A4

Two master files A and B are associated with one structure file Z. Links between records are set as indicated by the arrows in the diagram. The structure records carry the record addresses of the units they link, and not the unit names.

Notes: Data not directly related to this example has been omitted from the record diagrams.

Parents and subunits could of course be on the same file. An example where they are on different files has been taken for the sake of clarity.

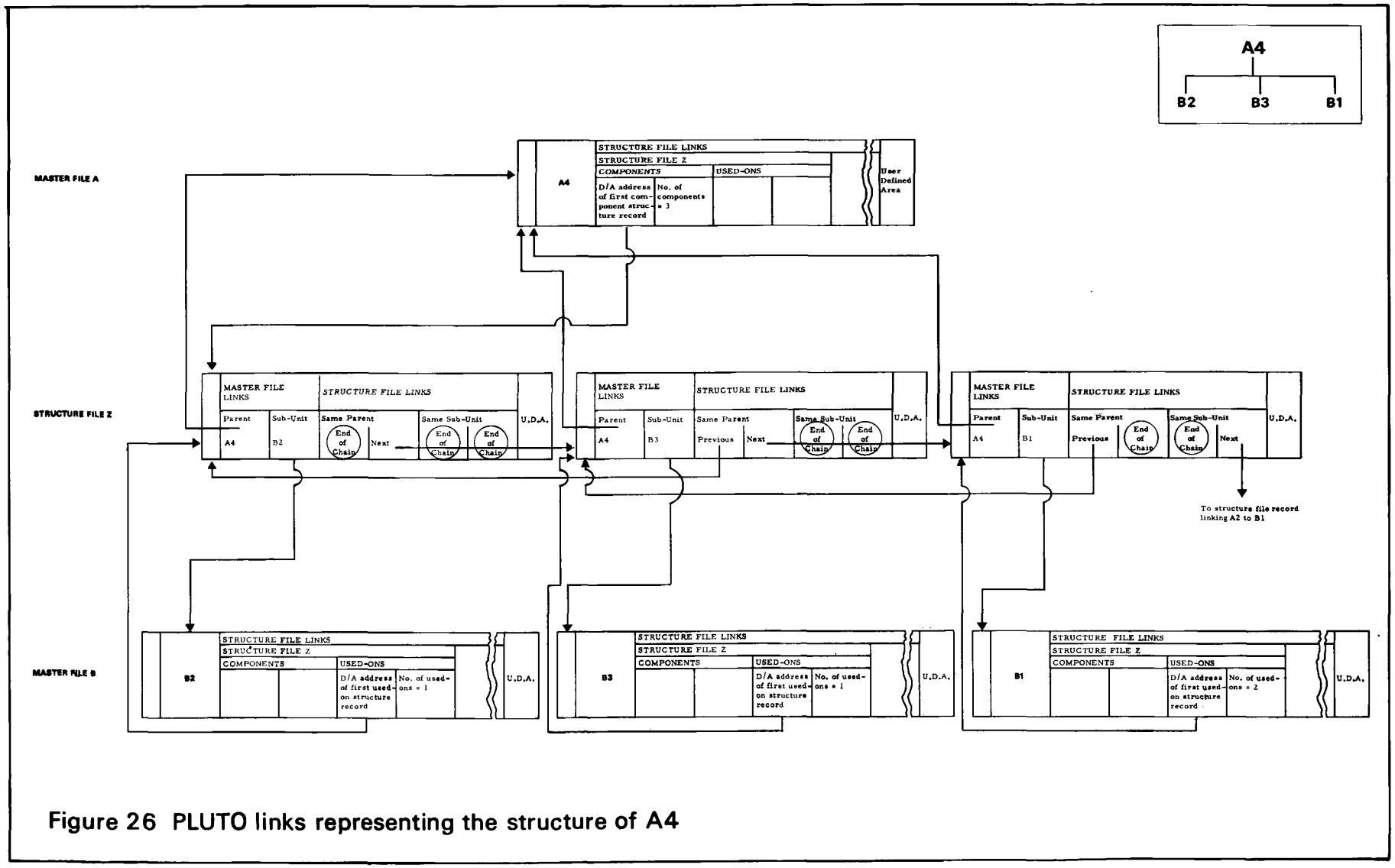


Figure 26 PLUTO links representing the structure of A4

In exploding A4, PLUTO starts at the master unit record for A4. From there it accesses the first component structure record (the record which links A4 to B2) which is the only such record that can be accessed directly (Figure 27).

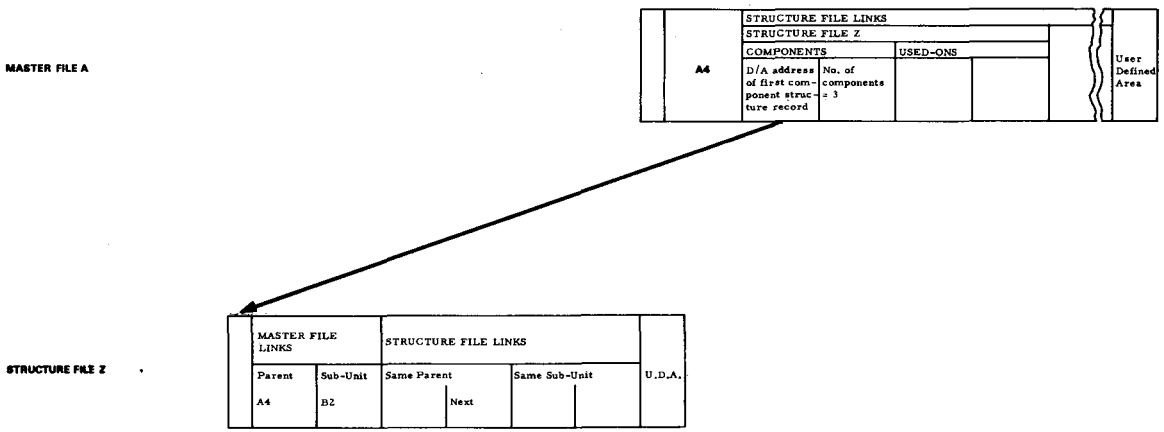


Figure 27 Link to first component structure record

The first component structure record carries the address of the master unit record for unit B2. The first component structure record is the first record in the structure chain which links all structure records having A4 as parent unit record.

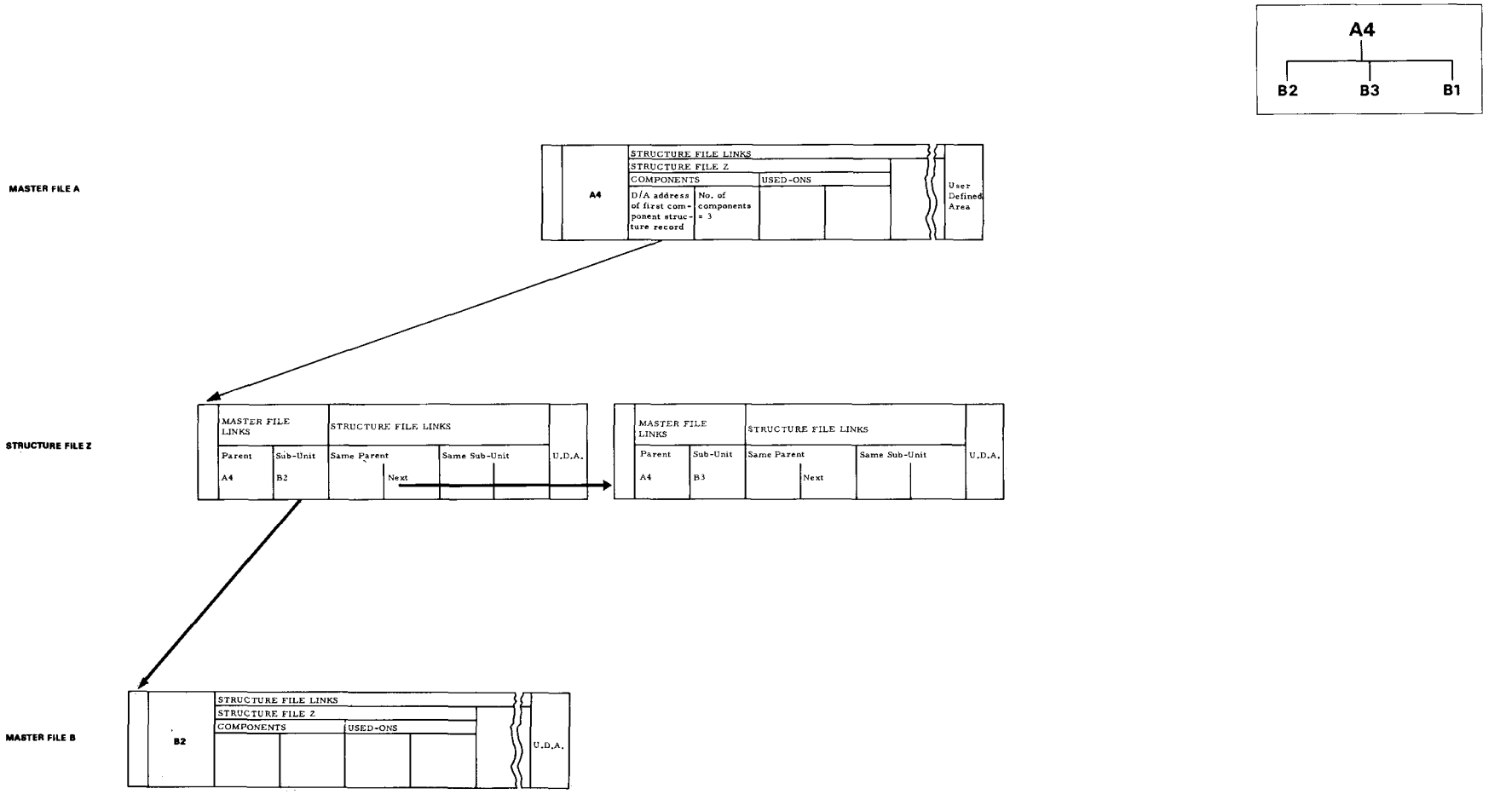


Figure 28 Links to subunit and second component structure record

Thus the first component structure record gives both the address of B2 and the address of the structure record which links A4 to B3 (Figure 28). In turn, this structure record gives the addresses of unit record B3 and of the third structure record on the chain (Figure 29).

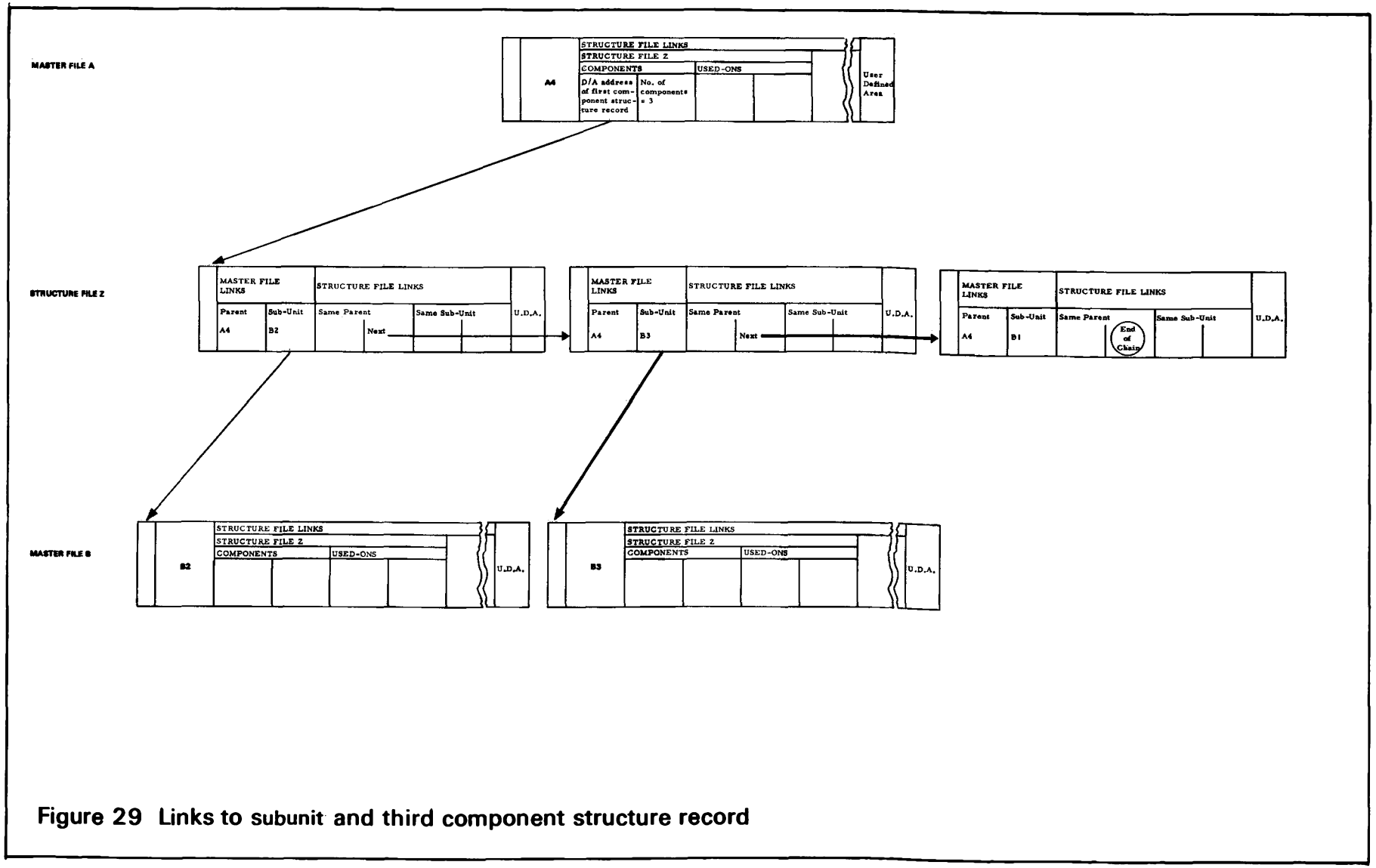


Figure 29 Links to subunit and third component structure record

The third structure record gives the address of the unit record for B1 and also carries an end of chain marker (Figure 30). The number of components is also given in the master unit record for A4, as a check.

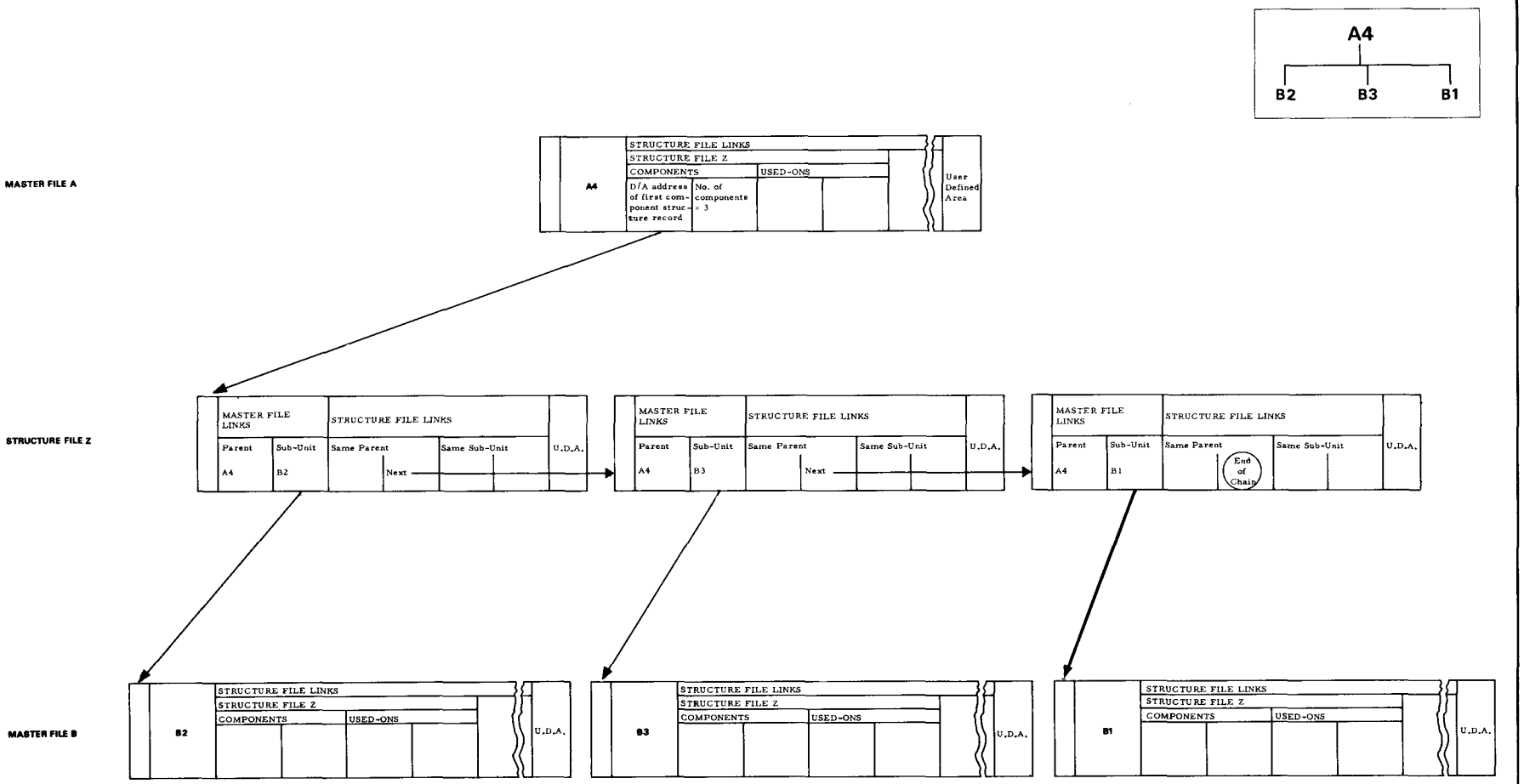


Figure 30 Link to subunit

The use of PLUTO links in implosion is identical to their use in explosion. Figure 31 shows the links followed in the implosion of B1. The master record for unit B1 gives the address of the first used-on structure record, which links B1 to A4. The first used-on structure record gives the address of its neighbour having the same subunit. This neighbour links B1 to A2, and carries a marker to indicate the end of the structure chain. The number of used-ons is also given in the master unit record for B1, as a check.

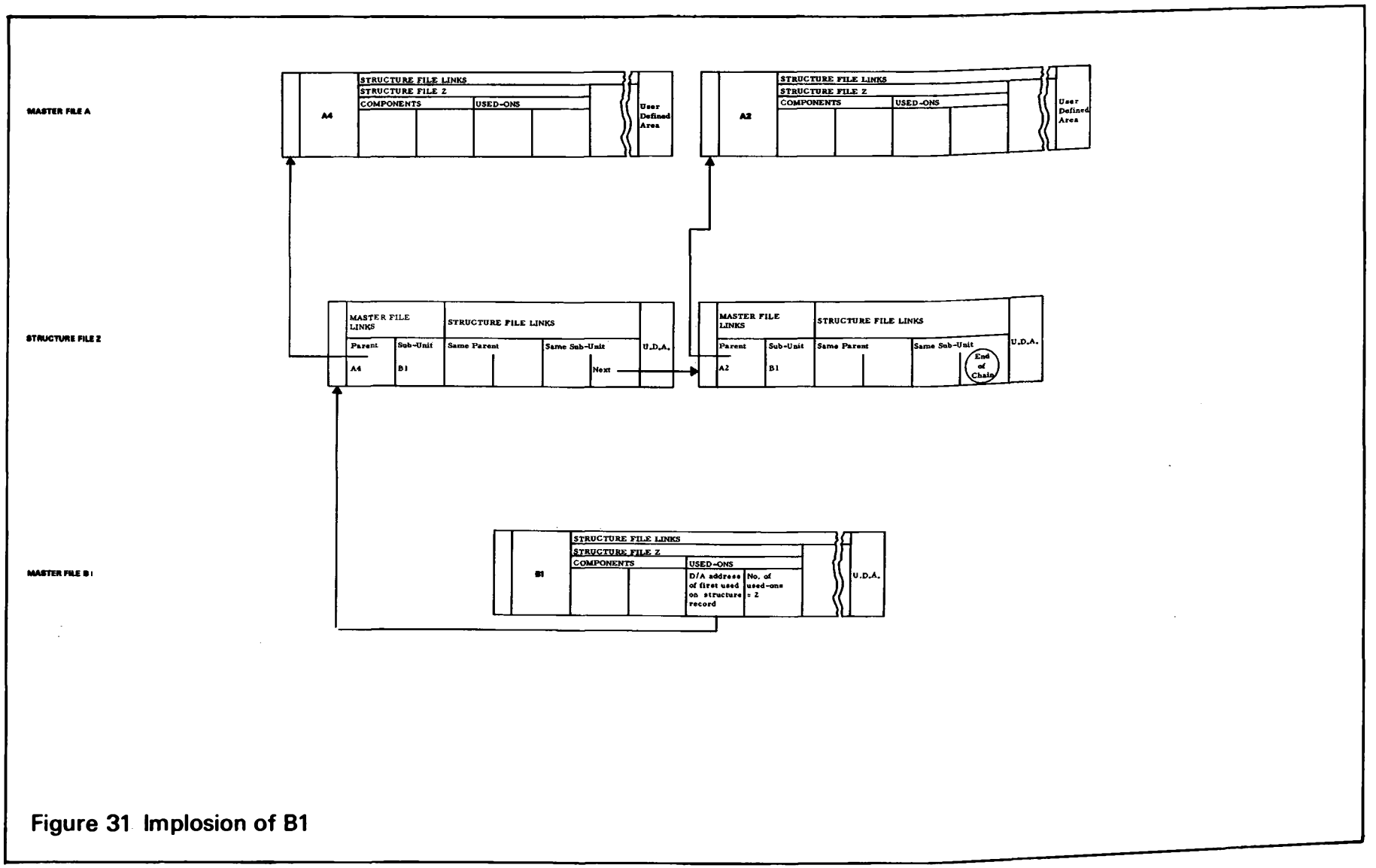


Figure 31 Implosion of B1

Condition codes

A structure record may be used to represent a conditional link. This is expressed by the use of condition codes which are held as a string of characters in the user defined area of the record. Continuation records may be used to hold excessively long condition codes. Several condition codes may be held in one record: either these may be used as a series of simple conditions or some or all may be grouped into compound conditions.

If a series of simple conditions is held, then if any one (or more) of these conditions is satisfied the link is used. If a series of condition codes forms a compound condition, then all of the simple conditions must be fulfilled before the link is used. There may be a Quantity Modifier in any condition (not more than one in any compound condition). The user supplies figures to be used with each quantity modifier. When a condition including a quantity modifier is fulfilled, the basic quantity, held in the PLUTO area of the record, is multiplied by the figure supplied by the user, to give the quantity of the subunit required per one of the parent unit.

The conditions are scanned serially either until one is satisfied, in which case the link is used, and the basic quantity is multiplied by the figure supplied with the quantity modifier, if there is one; or until the end of the series of conditions is reached, without any of the conditions being fulfilled, in which case the link is not used.

Use of testwords with quantity modifiers

Where the figure to be used to modify the basic quantity cannot be given globally for all records then a figure can be taken from any testword in the master record to multiply the product of the basic quantity and the figure associated with the modifier. The quantity modifier names the testword which is to be used.

PRESENTATION OF DATA

All the PLUTO file maintenance functions accept data from core. It is the user's responsibility to present the information in the required format (see the I.C.T. Manual *PLUTO Basic System).*

This gives the user the utmost flexibility in arranging input to the computer system. He does his own validation on data content, any sorts that are expedient and can of course use one source document to generate information to be placed on several files. Testwords are written by presenting testword number, security key and the information. Testwords are read by presenting testword number and security key. Other user fields are written by presenting field name and number of repeats together with the data. The number of repeats must be stated even if the field is non-repeatable and the number of repeats is zero. These fields are read by presenting field name and number of repeats. The order in which the field names are presented is at the user's discretion, but it is marginally more efficient for the sequence to be the same as that in which the fields were presented to the corresponding file creation run.

Chapter 5 Standard retrieval

This chapter explains how the user's program controls the standard explosion and implosion functions. These functions operate on one master file and one structure file at a time. If PLUTO finds links to unit records which are not on the master file which is currently being processed, tags may be output together with the other data retrieved. Control is returned to the user program, and the user must decide whether these tags are to be followed by a further call of the appropriate standard retrieval function.

PARAMETERS PRESENTED TO RETRIEVAL FUNCTIONS

When calling PLUTO standard retrieval functions, the user specifies the following:

1 The names of the master file and the structure file which are to be processed by this call of the function.

2 A list of names of those master files for which tags are to be output: PLUTO will not follow a reference to a unit record which is neither on the master file currently being processed, nor on a file whose name is on this list.

3 Lists of codes for conditional links may be presented. If the condition codes hold quantity modifiers, the user also presents the figures to be used with the modifiers, as explained in Chapter 4 (page 40 'Condition Codes').

4 Data to be exploded or imploded: this varies from function to function, and is described for each function below.

5 The output medium (E.D.S., F.D.S., or Magnetic Tape) and the name of the output file.

6 The data fields to be retrieved from the user-defined area of each record. A testword is specified by its number and its security key. Any other field is specified by its symbolic name.

Full details of the parameters required are given in the I.C.T. manual 'PLUTO Basic System'.

OUTPUT

PLUTO returns data and tags to backing store, which may be E.D.S., F.D.S., or Magnetic Tape.

Control is then returned to the user's program, and it is for the user to decide whether to follow tags by a further call of the appropriate PLUTO retrieval function, or to output tags, or to ignore them.

Output records

Full output records carry the following information:

1 Quantity.

2 Unit Name.

3 Contents of data fields requested by the user.

Tag records carry the following information:

1 Tag Marker.

2 File name: the name of the master file on which the unit record is to be found.

3 Quantity.

4 Bucket and record number of the unit record.

Use of output file

PLUTO leaves the user completely free to make what use he pleases of the output file. The user may output the data to any other medium, or use it for further processing within his own program, or present it again to the same PLUTO retrieval function with a different pair of master and structure files. In the latter case, the file is copied across to a new output file, until a tag to the master file now specified is hit. This tag is then exploded, and the data written to the new output file.

SINGLE LEVEL RETRIEVAL

The single level retrieval functions are 'PLUTO Single Level Explosion' and 'PLUTO Single Level Implosion'.

The user's program calls these functions presenting the parameters described above (page 4), 'Parameters presented to PLUTO retrieval functions').

Data to be exploded or imploded

The data to be exploded or imploded will be both

1 A quantity,

and

2 either a unit name,

or the direct access address (bucket and record number) of a unit record.

The unit will of course be on the master file which is currently being processed.

Explosion

PLUTO opens the files to be processed and reads into core the unit record to be exploded. The structure file link area on this unit record gives the address of the first component structure record. This record is read into core. It it is dead (i.e. if the erase bit has been set) the links and data fields are ignored and PLUTO reads into core the next component structure record, whose address is given in the first component structure record. If a structure record is live, PLUTO checks whether condition codes were presented by the user program; if there are conditions which must be satisfied, PLUTO compares the condition codes in the user-defined area of the structure record (Figure 19, page 20) with the list of condition codes presented by the user's program. If the conditions are not satisfied PLUTO ignores the links and data fields, and reads into core the next component structure record. If a condition is satsified, PLUTO checks whether there is a quantity modifier in the satisfied condition code and if so multiplies the basic quantity by the figure supplied by the user for use with the modifier. The product of this calculation (or the basic quantity, if no modifier was held) is multiplied by the quantity presented with the data to be exploded, to give the final quantity to be output.

PLUTO next reads from the master file link area the direct access address of the sub-unit's master record. If this record is on the master file which is currently being processed, PLUTO reads it into core and creates an output record containing all the data requested by the user's program. If the sub-unit's master record is not on the open master file but is on one of the files listed for which tags are to be output, a tag is output; otherwise there is no output. PLUTO then reads into core the next structure record on the component structure chain.

When all records on the component structure chain have been processed, control is returned to the user's program.

Implosion

'PLUTO Single Level Implosion' works in exactly the same way as 'PLUTO Single Level Explosion'. The unit record to be imploded is read into core; the used-on structure records are read into core one at a time. If a structure record is dead it is ignored. Condition codes (if input) are compared until a condition is satisfied and the quantity is then calculated. If the parent unit's master record is on the master file which is currently being processed, it is read into core and an output record is created containing all the data requested by the user. If the parent unit's master record is not on the open master file, but is on one of the master files for which tags are required, a tag is output. PLUTO then passes to the next structure record on the used-on structure chain. When all the structure records on the used-on chain have been processed, control is returned to the user program.

Example: see overleaf

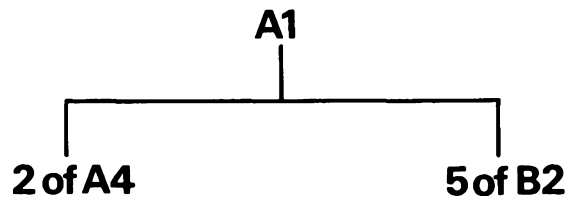


Figure 32 Structure of A1

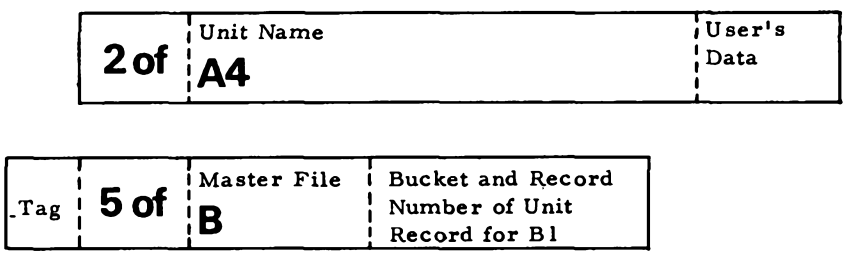


Figure 33 Output from first call of 'PLUTO single level explosion'

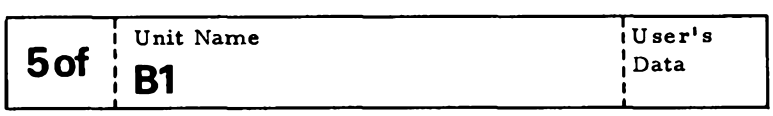


Figure 34 Output from second call of 'PLUTO single level explosion'

Example

Unit A1, whose structure is shown in Figure 32, has components on two different master files. Two calls of 'PLUTO Single Level Explosion' will be necessary to obtain the single level explosion of A1.

On the first call the user program will present the following parameters:

Master File A
Structure file (e.g.) Z } The pair of files to be processed

Tags are required to master file B

Data required from the user defined area is (e.g.) DES

Quantity (of A1) = 1

Unit name of unit to be exploded is A1.

and also specify where data is to be returned to. The output from this call will be a full record for A4 and a tag to master file B (see Figure 33).

The user program must then read the tag and call 'PLUTO Single Level Explosion' again, presenting the following parameters:

Master file B
Structure file Z } The pair of files to be processed

No tags are required

Data required from user defined area is DES

Quantity (as given in the tag) = 5

Bucket number and record number of the unit record for B1.

and also specify where data is to be returned to.

PLUTO will then output a full record for B1 (see Figure 34).

INDENTED RETRIEVAL

The indented retrieval functions are 'PLUTO Total Indented Explosion' and 'PLUTO Total Indented Implosion'.

The user's program calls these functions, presenting the parameters described above (page 41 'Parameters presented to PLUTO retrieval functions').

Data to be exploded or imploded

The data to be exploded or imploded may be either

1 The name of a unit or the direct access address of a unit record;

or

2 An input medium (E.D.S., F.D.S., or Magnetic Tape) and the name of the file produced by a previous call of the same function.

Explosion

When presented with a unit to be exploded, 'PLUTO Total Indented Explosion' scans the breakdown of the unit as completely as is possible using the current master file: before passing from one component of the unit to another, the first component is completely exploded. In indented retrieval, as may be seen from Chapter 1, page 6, the sequence of output is all important: tags, where required, are output in sequence. The user may present to a call of 'PLUTO Total Indented Explosion' either a single unit to be exploded, or a file which was produced by a previous call of 'PLUTO Total Indented

INDENTED EXPLOSION OF P1	OUTPUT FILE F PRODUCED BY FIRST CALL (Open Master File is P; Unit Name P1 is presented)	OUTPUT FILE G FROM SECOND CALL (Open Master File is Q; File F produced by First Call is presented)
7 of P2	Relative Level 1 | 7 of | Unit Name P2 | User's Data	Relative Level 1 | 7 of | Unit Name P2 | User's Data
1 of Q2	Tag | Relative Level 2 | 1 of | Master File Q | Bucket and Record No. of Unit Record for Q2	Relative Level 2 | 1 of | Unit Name Q2 | User's Data
1 of P3	Relative Level 2 | 1 of | Unit Name P3 | User's Data	Relative Level 2 | 1 of | Unit Name P3 | User's Data
2 of Q1	Tag | Relative Level 3 | 2 of | Master File Q | Bucket and Record No. of Unit Record for Q1	Relative Level 3 | 2of | Unit Name Q1 | User's Data
2 of P4	Relative Level 3 | 2 of | Unit Name P4 | User's Data	Relative Level 3 | 2 of | Unit Name P4 | User's Data
1 of P3	Relative Level 1 | 1 of | Unit Name P3 | User's Data	Relative Level 1 | 1 of | Unit Name P3 | User's Data
2 of Q1	Tag | Relative Level 2 | 2 of | Master File Q | Bucket and Record No. of Unit Record for Q1	Relative Level 2 | 2 of | Unit Name Q1 | User's Data
2 of P4	Relative Level 2 | 2 of | Unit Name P4 | User's Data	Relative Level 2 | 2 of | Unit Name P4 | User's Data
	This file is presented to the Second Call	

Figure 36 Total indented explosion of unit P1

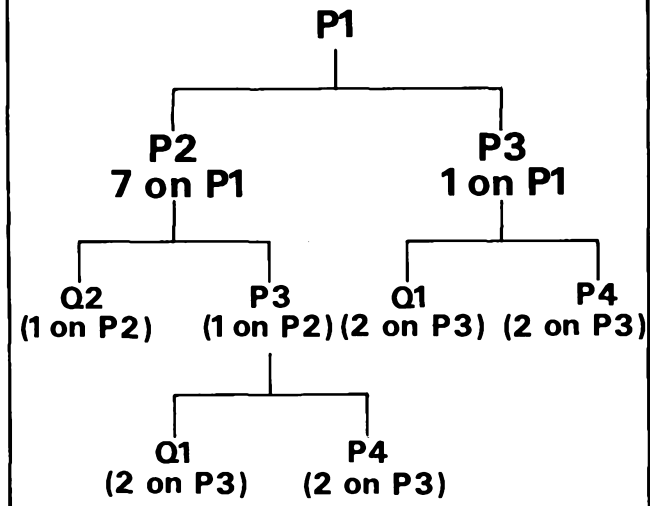


Figure 35
Structure of P1, showing quantities of direct components for all units

Explosion': in the latter case the contents of the file will be full output records mingled with unfollowed tags, some or all of which will reference unit records on the current master file. PLUTO copies the file across to a new output file until it hits a tag referring to the current master file; this tag is then exploded completely, and the full records are written to the output file in correct sequence. Both the full output records and the tags carry a relative level, indicating the degree of indenture.

Implosion

'PLUTO Total Indented Implosion' works in the same way as 'PLUTO Total Indented Explosion': the whole usage of one used-on of the unit presented is exploded before passing to the next used-on. If a file is presented, PLUTO copies it across, imploding tags to the current master file and writing the resulting records to the output file in correct sequence. The records produced also carry a relative level number.

Examples

Figure 35 shows the structure of unit P1. This unit has components on two master files, P and Q. To obtain the total indented explosion of P1, two calls of 'PLUTO Total Indented Explosion' are necessary.

On the first call, the user's program must present the following parameters:

Master file P
Structure file (e.g.) Z } The pair of files to be processed.

Tags are required for master file Q.

The unit name of the unit to be exploded is P1.

The output medium is (e.g.) E.D.S.; the output file name (e.g.) F.

The data required from the user-defined area is (e.g.) DAT

As shown in Figure 36 (centre column), PLUTO will write to output file F a mixture of full records for units which are on master file P and tags to unit records on master file Q, in appropriate order. The relative level number in the records indicates their level in the structure of the unit which is being exploded (P1 in this instance), as indicated by the indentation of the entries in the list on the left of Figure 36.

To retrieve the information for units which are on master file Q, the user's program must call the function again, presenting the following parameters:

Master file Q
Structure file (e.g.) Z } The pair of files to be processed.

No tags are required.

The input medium is E.D.S.; the name of the input file is F.

The output medium is (e.g.) E.D.S.; the name of the output file is (e.g.) G.

The data required from the user-defined area is (e.g.) DAT

The first record on file F is a full record, and it will be copied across to file G. The second record is a tag to master file Q: PLUTO will access the unit record whose direct access address is given in the tag, and create a full output record containing the relative level and quantity given in the tag, the unit name and the data requested by the user; this is the next record written to file G, as shown in the right-hand column of Figure 36. The third record is a full record which is copied across, the fourth record is a tag which is again exploded, and so on.

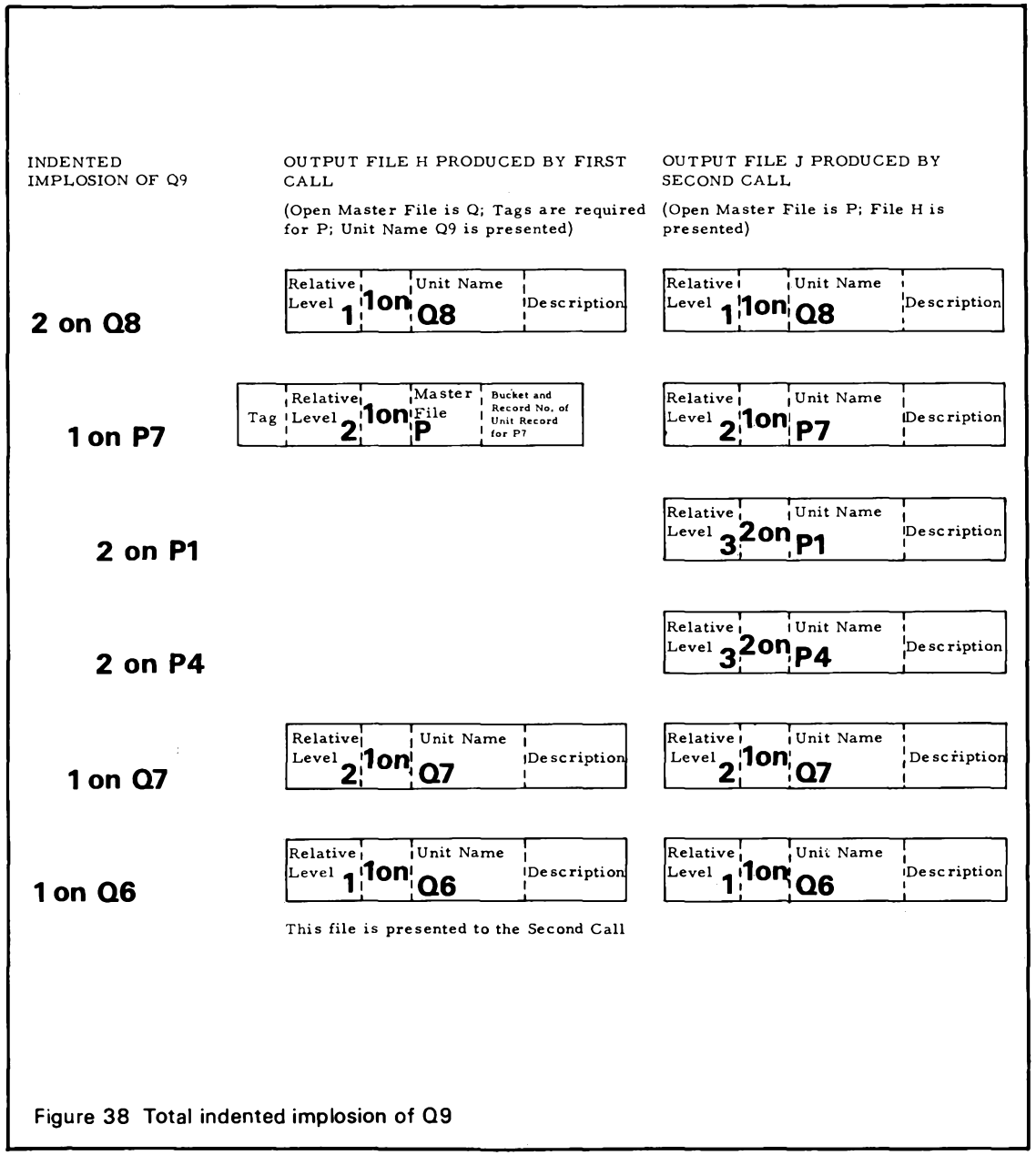


Figure 38 Total indented implosion of Q9

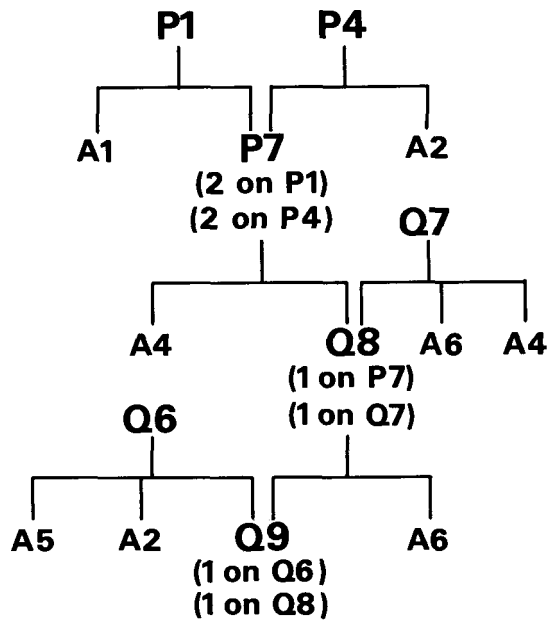


Figure 37 Structure showing usages of Q9

Where a tag refers to a unit which itself has a complex structure, the full exploration of this structure is inserted in the place which the tag occupied. Figure 38 illustrates this, taking as an example the total indented implosion of unit Q9, which forms part of the structure shown in Figure 37. Output file H, produced by the first call of 'PLUTO Total Indented Implosion' includes a tag to master file P amongst the full output records for units on master file Q. File H is presented to a second call of the function, with master file P open. The first record is copied across. The second record on file H is a tag to master file P; three records are produced as a result of the implosion of the tag, representing the full exploration of the usage of P7. These records are all written to file G before the third record from file H is copied across.

SUMMARIZED RETRIEVAL

The summarized retrieval functions are 'PLUTO Total Summarized Explosion' and 'PLUTO Total Summarized Implosion'.

The summarized retrieval functions work through the structure level by level, summing the quantities for each unit, and create one output record for each unit, containing the total quantity and whatever data the user requests from the user defined area.

The user's program calls these functions, presenting the parameters described above (page 41, 'Parameters presented to PLUTO retrieval functions').

Data to be exploded or imploded

The data to be exploded or imploded may be either

1 A quantity and either a unit name or the direct access address of a unit record;

or

2 An input medium (E.D.S., F.D.S., or Magnetic Tape) and the name of a file produced by a previous call of the same function.

Explosion

LOW LEVEL CODES AND ACTIVITY CHAINS

A unit may occur at more than one level of the structure. So that only one output record is produced for each unit, PLUTO does not create an output record for the unit until processing the lowest level at which the unit is found. This is achieved by means of the low level codes described in Chapter 4 page 29. PLUTO builds up a series of acitivity chains, which are chains linking all units on the current master file which have the same low level code and are active on the current call of 'PLUTO Total Summarized Explosion'. (An active record is one for which an output record is to be created.) There can be one activity chain for each level, up to a total of 33 levels. Thus each unit is put on the activity chain for the lowest level at which it is found. The address of the first record on each activity chain is held in a table in core.

An activity chain table is set up in core for each master file called by the user program, by the PLUTO file initialization routine (see the I.C.T. manual 'PLUTO Basic System').

EXPLOSION

Given a unit for summarized explosion, PLUTO

1 creates an output record for the unit;

2 explodes the unit, and

   (a) calculates the quantity required for each subunit; this is the product of the quantity held in the counter of the parent's unit record and the quantity given by the structure record (basic quantity or basic quantity modified);

   (b) puts each subunit which is on the current master file onto the appropriate activity chain and writes the quantity required to the counter of the subunit's unit record;

   (c) outputs tags for subunits which are not on the current master file but are on a master file for which tags are required; the tags carry the quantity, file name, and direct access address of the unit record.

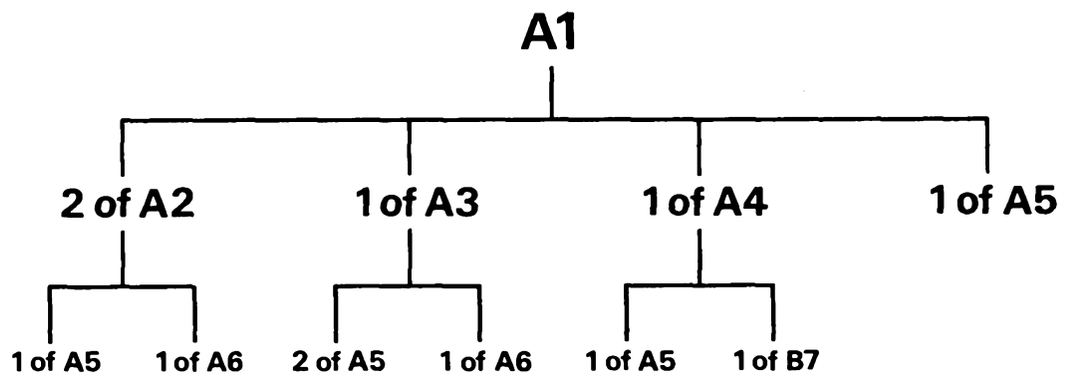


Figure 39 Structure of A1, with quantities

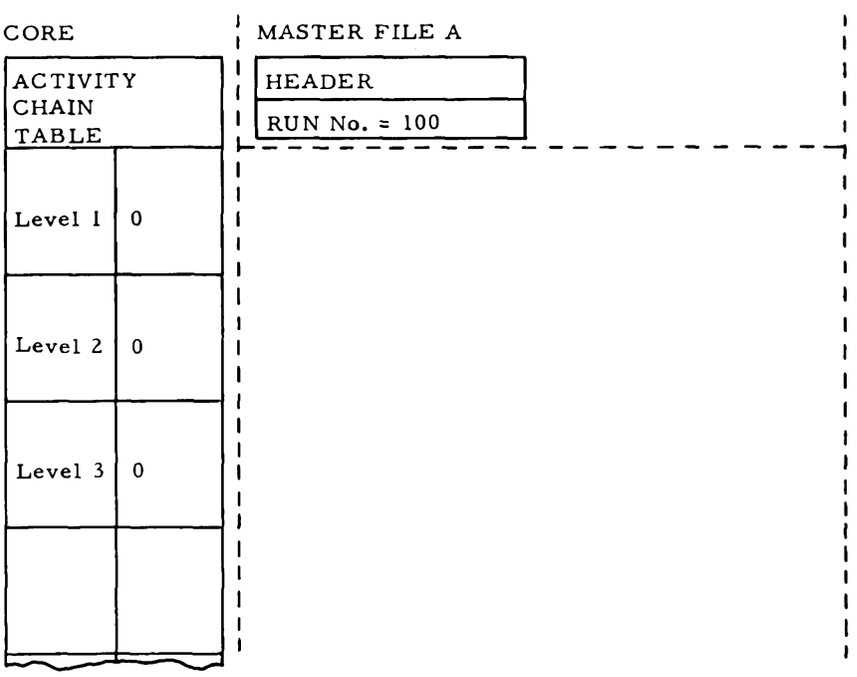


Figure 40 PLUTO updates run number and sets zeros in activity chain table

PLUTO then works along the activity chain for the next level beginning with the record whose address is in core. For each unit record on the chain an output record is created. PLUTO then explodes the unit, calculating quantities for subunits, putting the subunits on activity chains if they are on the current master file, and outputting tags (where required) if they are not. When this level has been completely processed, the activity chain for the next level is processed in the same way, and so on to the bottom of the structure.

Note: If the input is a file, PLUTO puts the units which are to be exploded onto chains and then processes the highest level activity chain, creating output records and exploding the units.

Implosion

'PLUTO Total Summarized Implosion' works in the same manner as 'PLUTO Total Summarized Explosion', except that it implodes units and puts parents on activity chains. The system of low level codes works for implosion as well as for explosion, since a unit will always have a lower low-level code than its used-ons. The lowest level activity chain is processed first.

Run numbers

For activity chains to work it is essential that a record should be put on an activity chain not more than once; summarized retrieval also requires that the total quantity of the unit be accumulated in the counter of the unit record. Therefore the first time a unit is accessed it must be put on an activity chain, and the quantity must be *written* to the counter, thus cancelling any quantity created on a previous run; on any subsequent access it must not be put on an activity chain, and the quantity must be *added* to the counter. These objectives are achieved economically by means of the system of Run Numbers. Each time a master file is used for summarized retrieval, the run number in the master file header is increased by 1. Each unit record holds the run number of the last run on which it was accessed. By testing whether the run number held in the unit record is equal to that in the master file header, PLUTO checks whether the unit record has been accessed previously on the current run: if so, it is already on an activity chain, and the quantity must be added to the counter; if not, it must be put on the activity chain, the run number in the master file header must be written to the unit record, and the quantity must be written to the counter.

Example

Total summarized explosion is required of a unit A1, whose structure is shown in Figure 39. All components of A1 are either on the same master file A, or on another master file B, (as indicated here by the letter A or B in the unit name).

No conditions are required.

The user calls 'PLUTO Total Summarized Explosion', presenting the following parameters:

Master file A
Structure file (e.g.) Z } The pair of files to be processed

Tags are required for master file B.

Zeroes to indicate 'no conditions'.

The unit name of the unit to be exploded is A1 and the quantity is 1.

The output medium is (e.g.) Magnetic Tape; the name of the output file is (e.g.) F.

The data required from the user defined area is (e.g.) INF.

PLUTO opens the files to be processed, and updates the run number in the header of master file A, making the current run number (e.g.) 100. PLUTO then accesses the activity chain table in core, to which activity chains are anchored: when activity chains are created, the direct access address of the first unit record on the chain for each level is held in this table. PLUTO begins by setting all entries in the activity chain table to zero (see Figure 40).

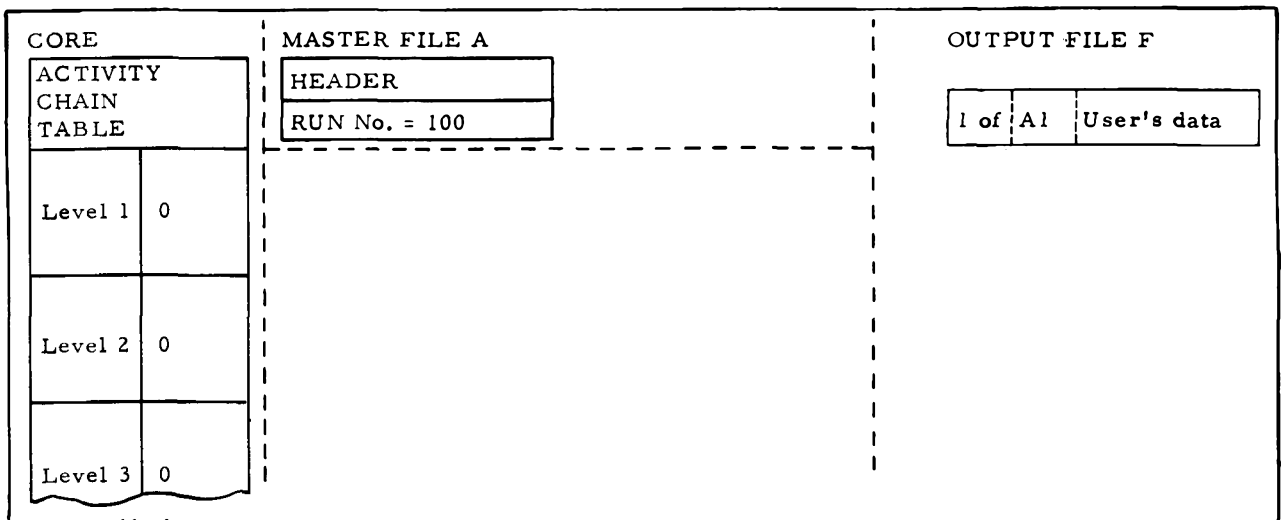


Figure 41 An output record is created for A1

MASTER FILE LINKS		STRUCTURE CHAIN LINKS				Basic Quantity
		SR's Having Same Parent		SR's Having Same Sub-Unit		
Parent	Sub-Unit	Previous	Next	Previous	Next	
A1	A2	0	A1 - A3	0	0	2

Figure 42 Structure record (A1-A2)

UNIT RECORD HEADER					Unit Name	Links to Structure Files	
Activity Chain		Count	Run Number	Low Level Code		Links to SF Z	
Next Record	Compare					First Component	First Used-On
D/A Address of A8	portion for last record	26	98	2	A2	D/A address of (A2 - A5)	D/A address of (A1 - A2)

Figure 43 Unit record for A2

The master file unit record for unit A1 is then read into core.

Output: An output record is created for A1, containing the quantity presented by the user's program, the unit name, and data requested by the user program (see Figure 41).

Explosion: PLUTO then explodes A1. The first component structure record for A1 is read into core. This is the structure record linking A1 to A2, whose address is given in A1 (see Figure 42). PLUTO checks condition codes: no conditions have been input. PLUTO calculates the quantity of A2 required by multiplying the basic quantity given in the structure record (see Figure 42) by the quantity given for the parent unit. This is 2×1 = 2. PLUTO checks whether the subunit is on the current master file. It is, so the subunit's unit record is read into core. This might be as shown in Figure 43.

PLUTO checks the run number of the subunit's unit record for equality with the run number in the file header: they are not equal, so the unit record has not been accessed on this run.

PLUTO updates the run number in the unit record by writing the number from the file header to the unit record.

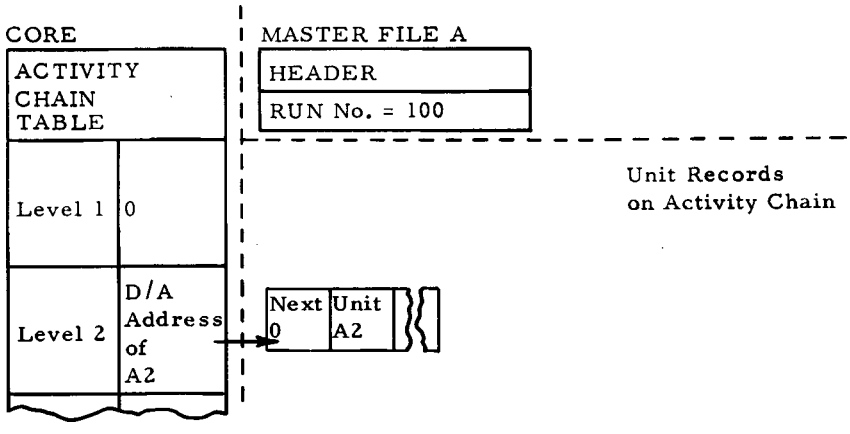
PLUTO checks the low level code of the unit. This is 2 so the record should be put on the activity chain for the second level. PLUTO reads the present contents of the activity chain table level 2 and enters them in the unit record as the address of the next record on the activity chain. In this case, the activity chain table level 2 holds zero (see Figure 40) so zero is entered, which indicates the end of the activity chain. The direct access address on master file A of unit record A2 is then entered in the activity chain table, level 2.

PLUTO writes to the counter of the unit record in core the quantity (2) previously calculated.

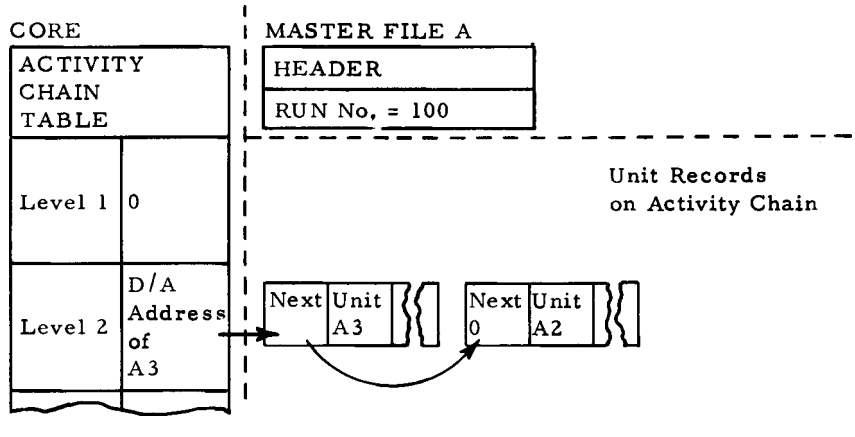
The updated record (see Figure 44) is written back to the file.

UNIT RECORD HEADER					Unit Name	Links to Structure Files	
Activity Chain		Count	Run Number	Low Level Run		Links to SF Z	
Next Record	Compare portion for last record					First Component	First Used-On
0		2	100	2	A2	D/A Address of (A2 - A5)	D/A Address of (A1 - A2)

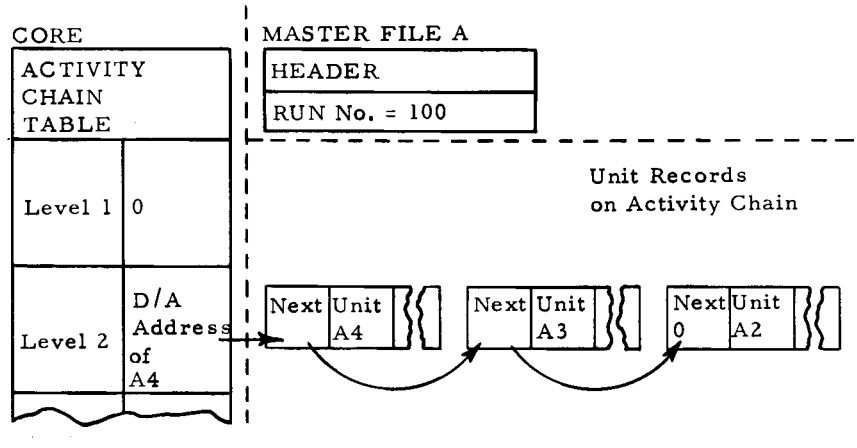
Figure 44 Updated unit record for A2



(i) PLUTO reads zero (end of chain) from table and puts it in A2; PLUTO puts address of A2 in table.



(ii) PLUTO reads address of A2 from table and puts it in A3; PLUTO puts address of A3 in table.



(iii) PLUTO reads address of A3 from table and puts it in A4; PLUTO puts address of A4 in table.

Figure 45
Building up of the activity chain (level 2). The unit records on the completed chain are shown in Figure 46

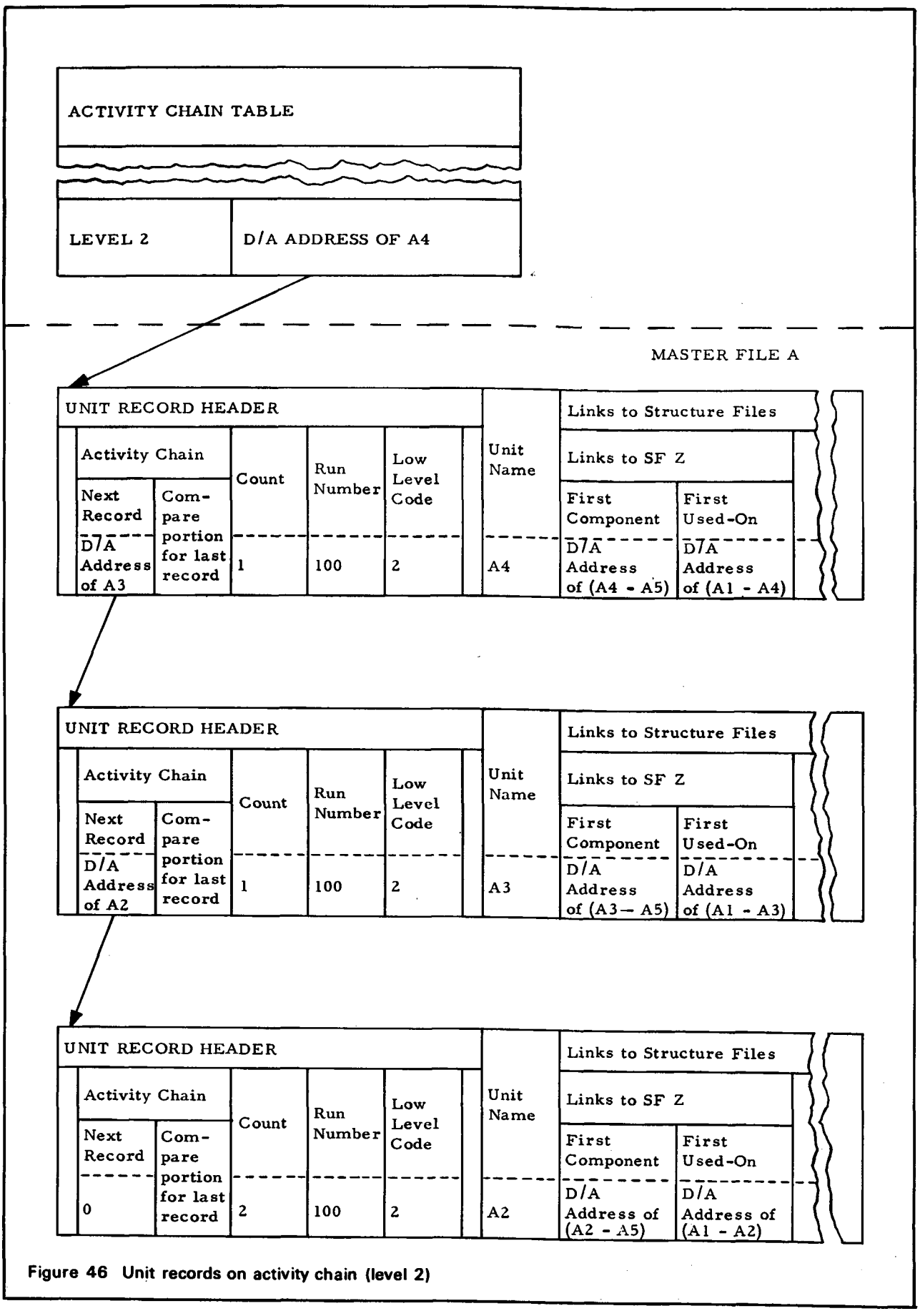


Figure 46 Unit records on activity chain (level 2)

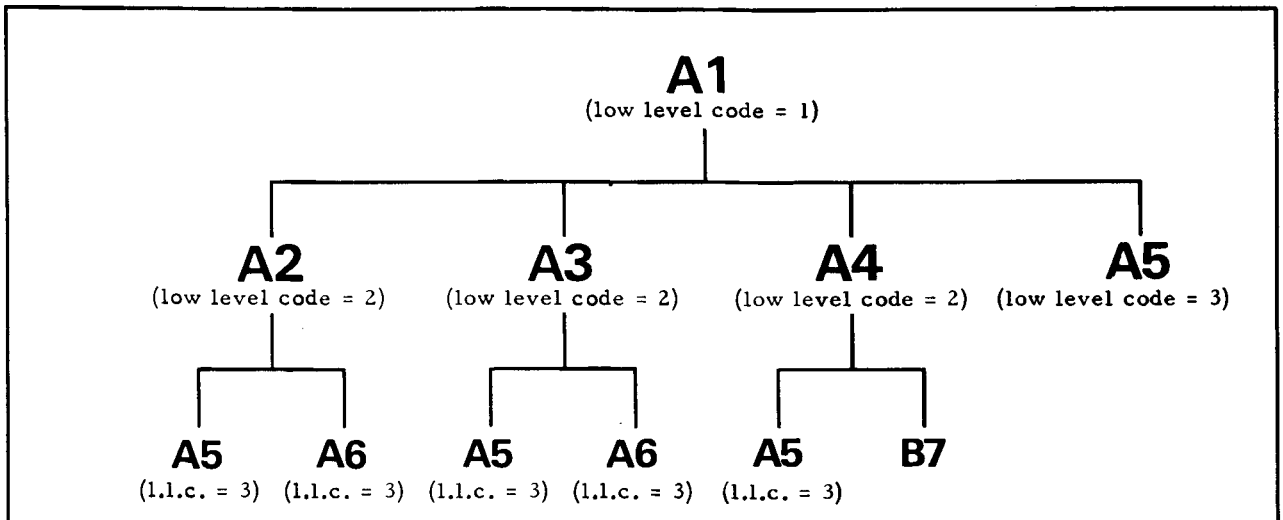


Figure 47 Structure of A1 showing low level codes for units on master file A

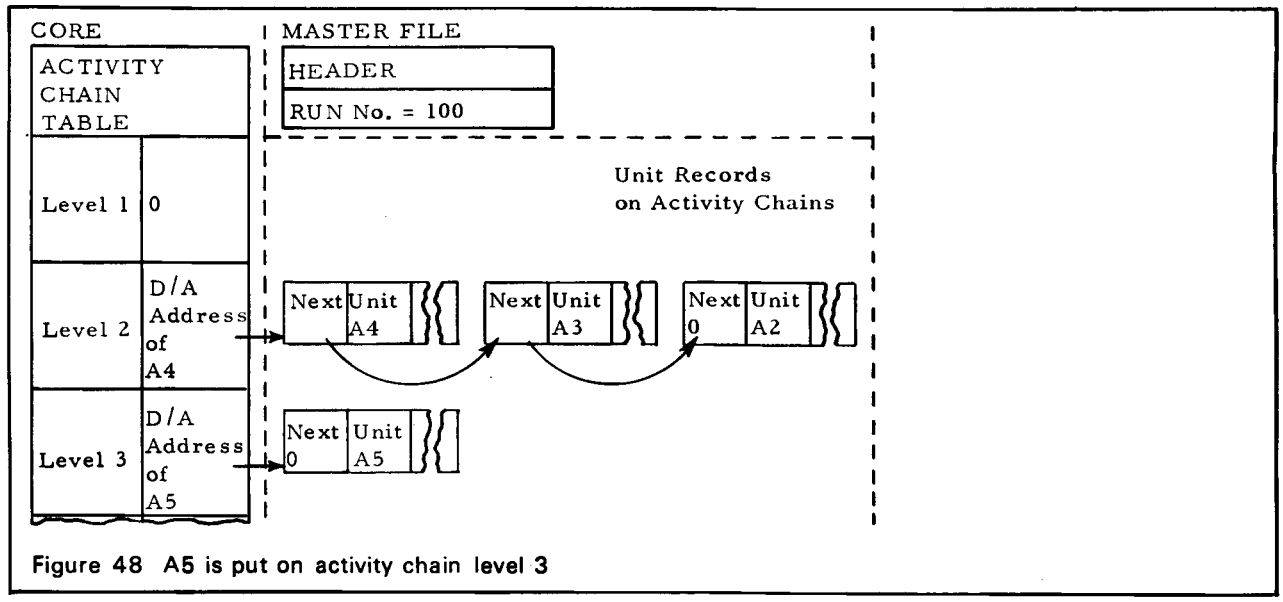


Figure 48 A5 is put on activity chain level 3

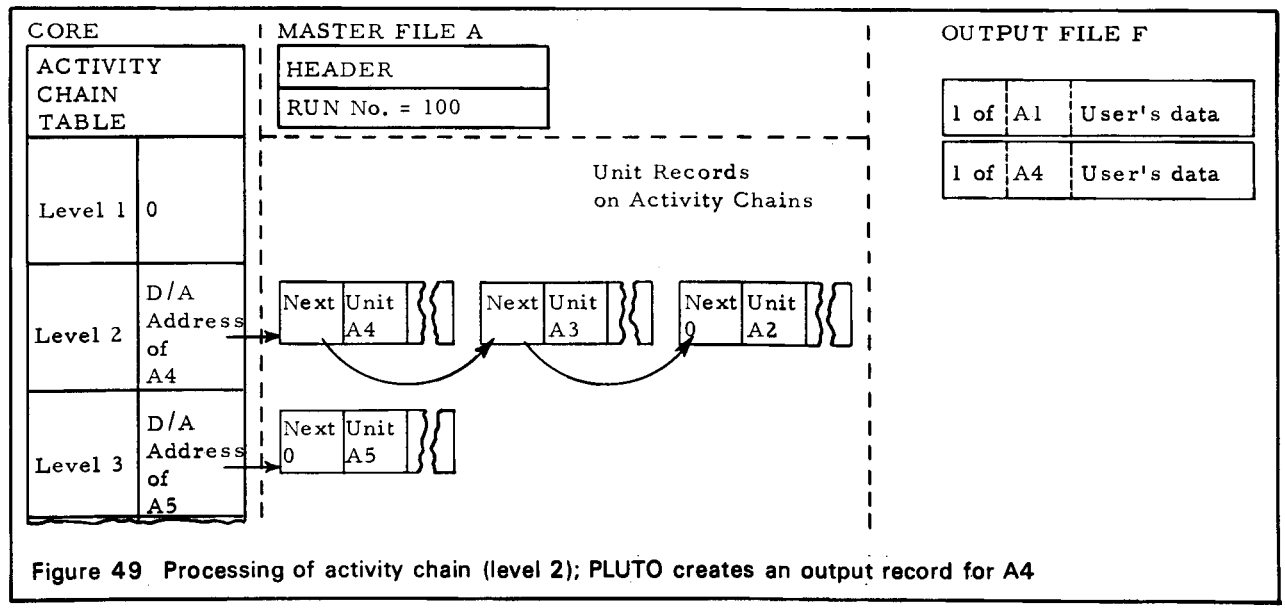


Figure 49 Processing of activity chain (level 2); PLUTO creates an output record for A4

PLUTO proceeds along the structure chain continuing the explosion of A1, updating the component records, and building up the activity chain for level 2, as shown in Figures 45 and 46. The last record in the structure chain links A1 to A5. Since A5 is also a component of A2, A3 and A4, it has a low level code of 3 (see Figure 47), and is therefore put on the activity chain for level 3 (see Figure 48).

Note that the last record to be added to the activity chain becomes the first on the chain, and will be the first to be processed, thus A4 will be the first record on the activity chain for level 2 (see Figure 49).

When there are no more records on the structure chain, PLUTO passes to the next level.

PLUTO reads in the first record on the activity chain level 2 (i.e. A4).

PLUTO creates an output record for A4 (see Figure 49).

PLUTO then explodes A4. The first component structure record links A4 to A5. A5 is on the current master file, so its unit record is read into core. The run number shows that A5 has already been accessed on this run, and has therefore been placed on the activity chain for the appropriate level, and holds a quantity in the counter which refers to this run. The quantity, calculated by multiplying the basic quantity (given in the structure record linking A4 to A5) by the quantity for A4 (given in the counter of the unit record for A4), is therefore *added* to the counter of A5. The unit record for A5 is then written back to the master file. PLUTO proceeds to the next record on the structure chain: this links A4 to B7. The quantity is calculated in the same manner as before. B7 is not on the current master file, so PLUTO checks whether tags are required for master file B: they are, so a tag is output (see Figure 50).

There are no more component structure records for A4, so PLUTO reads in the next record on the activity chain, whose address is given in A4, (i.e. A3), creates an output record for A3, and explodes A3. The first component of A3 is A5, which is already on an activity chain, so the quantity is added to the counter of A5 and it is written back to the master file. The second component is A6, which has not been accessed on this run, so it is put on the activity chain (level 3) and the quantity is written to the counter. The situation is now as shown in Figure 51.

There are no more components of A3 so PLUTO passes to the next record on the activity chain, i.e. A2, creates an output record, and explodes it as before, calculating the quantities, and accumulating them in the counters of the subunits' master records.

A2 carries a zero in the 'next record on the activity chain' field, indicating the end of the activity chain (level 2), so PLUTO passes to level 3. A6 is read into core and an output record created; it has no components. A5 is read into core and an output record created; it has no components, and carries the end-of-activity-chain marker, so PLUTO passes to the next level in the activity chain table; this contains zero, and so do all the remaining levels indicating that there are no more activity chains to be processed. PLUTO returns control to the user's program. The final stage is illustrated in Figure 52. The user's program may sort or otherwise process the output file, and must call PLUTO again if it is desired to follow the tag to master file B.

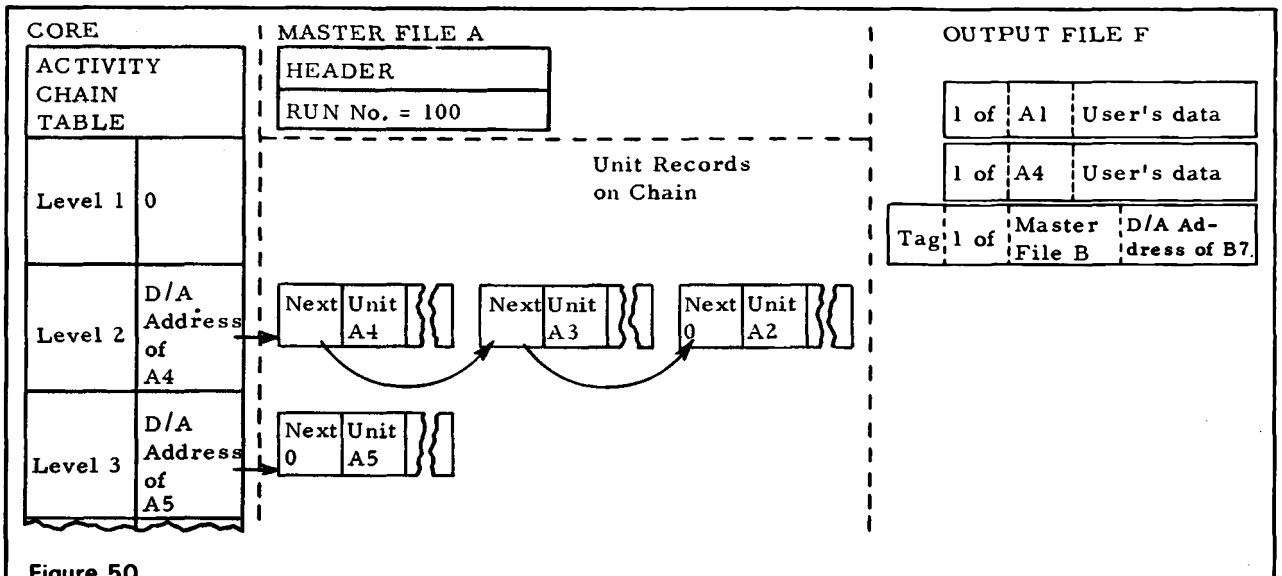


Figure 50
Processing of activity chain (level 2); a tag is output for a component of A4 not on the current master file

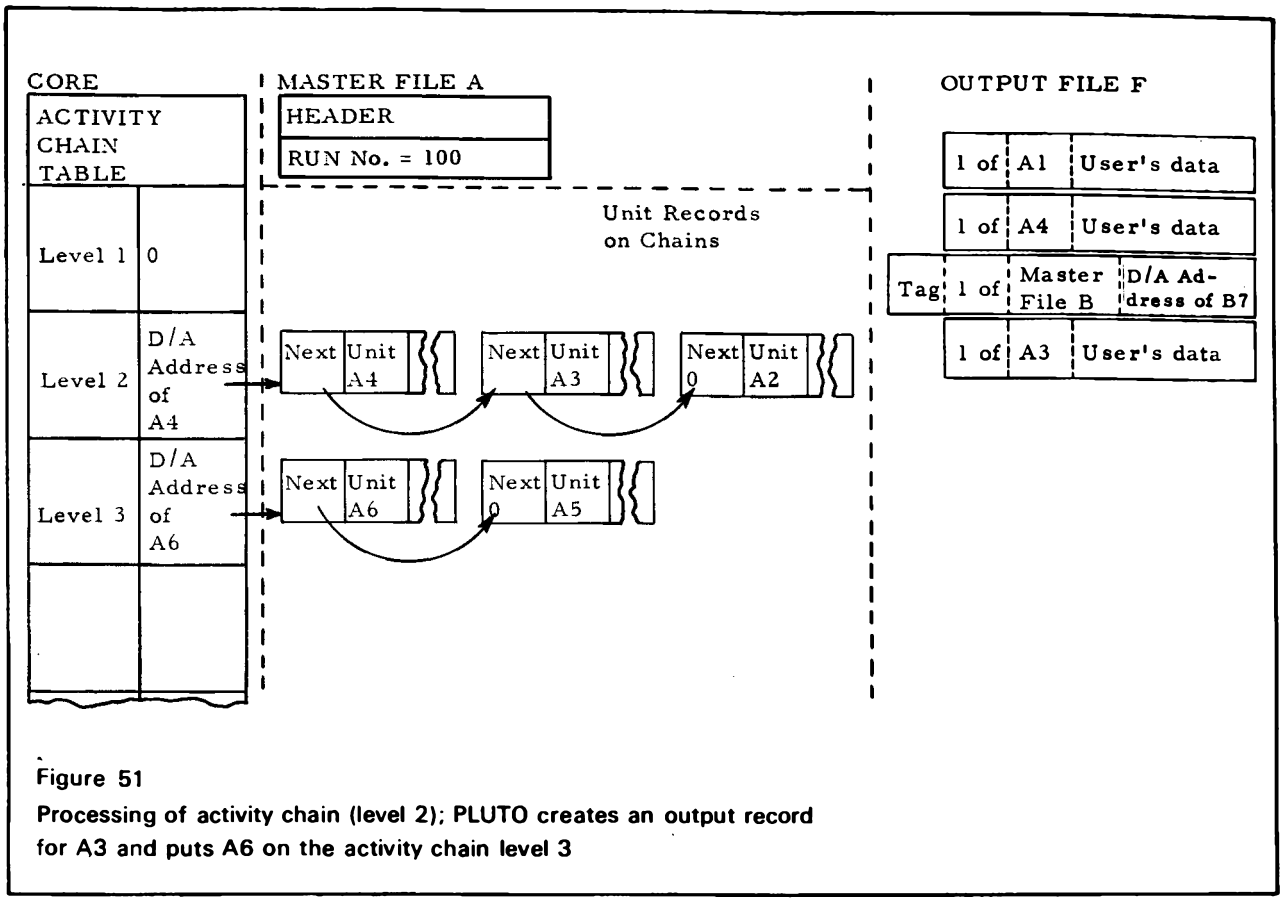


Figure 51
Processing of activity chain (level 2); PLUTO creates an output record for A3 and puts A6 on the activity chain level 3

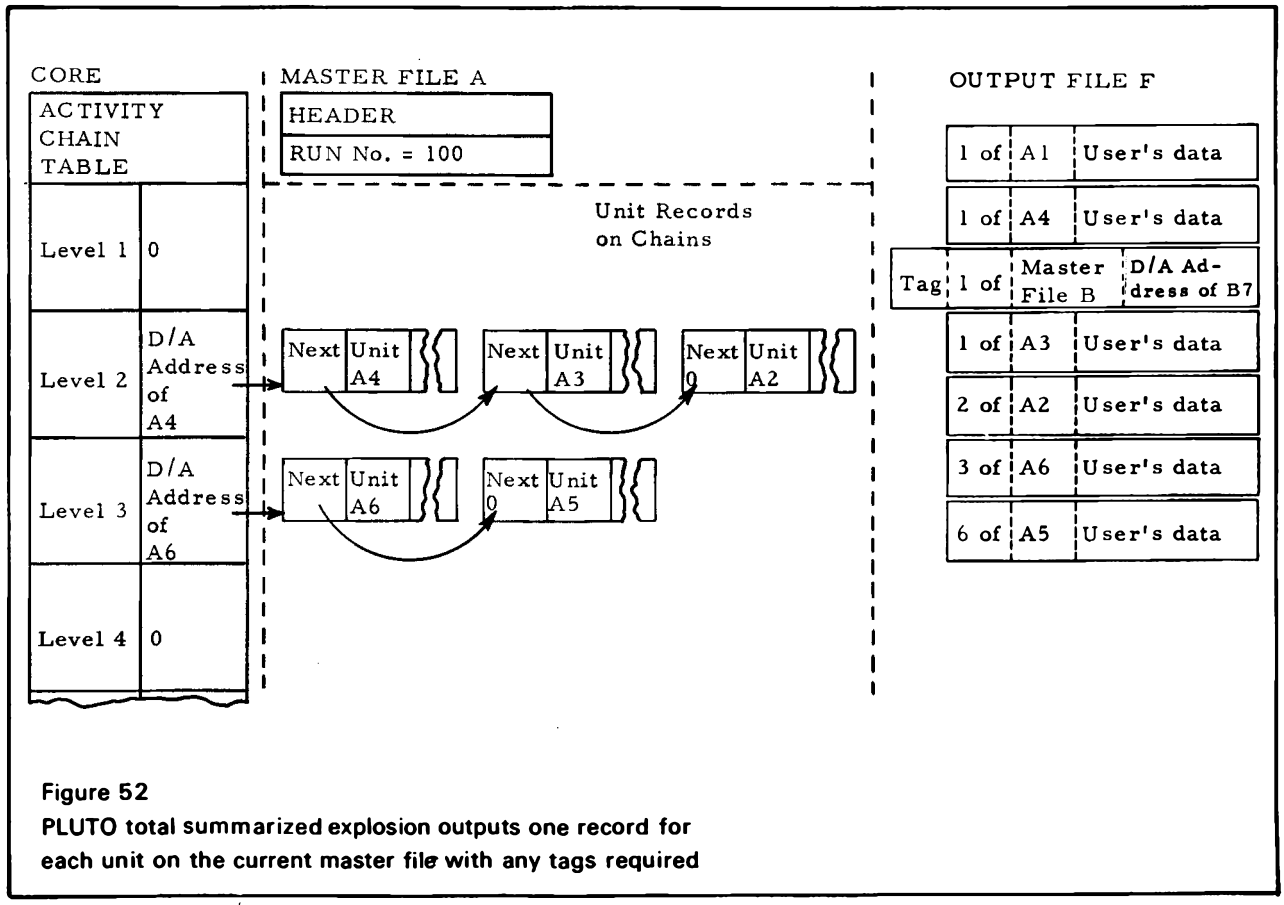


Figure 52
PLUTO total summarized explosion outputs one record for each unit on the current master file with any tags required

Chapter 6 PLUTO basic functions

Chapter 5 dealt with the PLUTO standard retrieval functions. The user is not limited to these, however, and may write his own programs using the PLUTO basic functions. These functions perform the individual steps that are necessary for whatever data handling and retrieval the user requires. Using these as basic bricks the user can build his own routines for non-standard forms of retrieval (e.g. to put lead times on indented explosion, or to retrieve data from the user defined area of structure records). The functions are described here under four headings, which classify the four types of elementary function which the user will need: basic record handling and retrieval functions, quantity and condition functions, chain functions, and basic structure exploration functions. Full details are given in the I.C.T. manual "PLUTO Basic System".

BASIC RECORD HANDLING AND RETRIEVAL

These functions serve the purposes of a conventional disc housekeeping system. They give access to buckets and records, search indexes to obtain record addresses, and access fields within records. The bucket and record accessing functions are deceptively simple: PLUTO automatically opens the file if necessary and finds any necessary file headers and layout strings.

Given the name of a unit, PLUTO will search the indexes and read into core the bucket and record address of the appropriate unit record.

Given the address of a bucket or record on a file, PLUTO will read the bucket or record into core.

PLUTO basic functions also enable the user to deposit in core the unit record which occupies a given position within a group; and to read a specified field from a group header.

The contents of a particular field within a unit or structure record can be retrieved when the record is in core. Such fields are (in the user defined area) unit name, testwords, unit and group dependent information, and user's structure record information. Testwords may be updated in core. Fields in the PLUTO area may be read and updated (see "Quantity and Condition Functions" below). Updated unit record and buckets may be written back to PLUTO master files.

QUANTITY AND CONDITION FUNCTIONS

Unit records

The system of run numbers described in Chapter 5 pages 49 to 57 ("Summarized Retrieval") is also available for the user's own retrieval programs.

RUN NUMBER

PLUTO basic functions enable the user to read and (where required) to update the run number on a unit record which is in core.

COUNTER

PLUTO basic functions enable the user the read the counter on a unit record which is in core, and to write or add a quantity to the counter on a unit record which is in core.

Note: The user calls the PLUTO basic 'write record' or 'write bucket' function to write away the updated record (see "Basic Record Handling" above).

Structure records

Given a structure record in core, PLUTO basic functions enable the user to read the basic quantity and to read condition codes.

CHAIN FUNCTIONS

Activity chains

The system of activity chains described in Chapter 5 pages 49 to 57 ("Summarized Retrieval") is also available for the user's own retrieval programs.

PLUTO basic functions enable the user to put records on activity chains, and to retrieve successively the records on an activity chain.

Level chains

PLUTO also maintains level chains, which link together all units having the same low-level code. These chains are set up in the parent master file by the PLUTO structure file loading function (see Chapter 4 pages 27 to 29, "Loading PLUTO Files"), and are available for the user's own retrieval programs.

Given a unit record in core, PLUTO basic functions will read into core the previous or the next record on the same level chain.

BASIC STRUCTURE EXPLORATION

PLUTO basic structure exploration functions enable the user to perform the following operations:

1 to find the number of component or used-on structure records linked to a unit record which is in core;

2 to read into core all the component structure records or all the used-on structure records linked to a given unit record;

3 to read into core the first component structure record or the first used-on structure record linked to a unit record which is in core;

4 given a structure record in core, to read into core the next or previous structure record on the component or used-on structure chain;

5 given a structure record in core, to read into core its parent's or its subunit's master record.

THE USE OF BASIC FUNCTIONS

The user can build up any retrieval routine he requires, from the PLUTO basic functions. This section gives two simple examples. The first shows how the user could build up his own routine to do the work of Single Level Explosion. The second shows how data could be retrieved from a sequence of structure records, as required, for example, in the Routings application. Full details of the functions employed are given in the I.C.T. Manual "PLUTO Basic System".

A single level explosion routine

Unit name converted to record number: the user's program presents the unit name, and PLUTO searches the indexes and reads the direct access address of the unit record into a core location specified by the user.

Read record: the user's program presents the address of the unit record, and PLUTO returns the record required to an area in core specified by the user.

Read first structure record: the user's program presents the unit record which is in core, and PLUTO returns its first component structure record to a core location specified by the user.

The structure record is then processed as follows:

Read codes in structure: PLUTO moves the condition codes to a core location specified by the user.

Own coding: these codes are compared with codes presented by the user. If none of the conditions is satisfied, the link is ignored, and the next structure record on the chain is read in and processed. If a condition is satisfied, the program checks whether the subunit is on the current master file, and if not, whether it is on the list of files for which tags are to be output. If a tag is not required, the next structure record on the chain is read in and processed.

Read basic quantity: the basic quantity in the structure record is read into an area specified by the user.

Own coding: the program checks whether a quantity modifier is present in the satisfied condition, and if so multiplies the basic quantity by the figure supplied for use with the quantity modifier. If the subunit is not on the current master file the program outputs a tag and the next structure record is processed.

Read component unit: the user program presents the structure record in core and PLUTO returns the subunit's master record to a core location specified by the user.

Read unit name/read test word/read unit information/read group dependent unit information: using these functions, the user retrieves the data necessary to create the output record he requires.

The output record is written away, and the next structure record on the chain is read into core. The routine is repeated until all records on the structure chain have been exploded. Control is then returned to the main program.

A routings retrieval routine

The user could build up a routine to retrieve data from a sequence of structure records, representing operations, as follows:

Unit name converted to record number: the user's program presents the name of the part whose routing is required, and PLUTO searches the indexes and returns the direct access address of the unit record to a core location specified by the user.

Read record: the user's program presents the direct access address of the record, and PLUTO returns the record to a core location specified by the user.

Read test word/read unit information/read group dependent information: by means of these basic functions, the user retrieves the data he requires from the part unit record.

Own coding: the user's program creates an output record for the part, carrying whatever data the user requires.

Read first structure record: the user's program presents the part unit record in core, and PLUTO returns to a specified core location the first component structure record, i.e. the record for operation number 1.

Read test word in structure record/read user's structure information: by means of these basic functions, the user retrieves from the user defined area of the structure record the required data relating to the operation.

Own coding: the user's program creates an output record for the operation. As well as the operation number and the operation data from the user defined area, this could carry the direct access address of a unit record on the work-centre master file. The user would build up a routine to follow these references from PLUTO basic functions.

Read next component: the user presents the structure record which is in core, and PLUTO returns to a specified location the next structure record on the component structure chain, i.e. the record for operation number 2. This is processed in the same way as the first structure record. The cycle is repeated until the end of the component structure chain is reached. Control is then returned to the main program.

Chapter 7 File reorganization

PLUTO File Reorganization is a free-standing suite of programs. This chapter outlines the uses of PLUTO File Reorganization, and the principles employed. Full details will be given in the I.C.T. manual "PLUTO Extended System".

USES OF REORGANIZATION

There are a number of reasons why, from time to time, a user may wish to reorganize a PLUTO file. He may wish to change the system of files, possibly introducing fresh structure or master files that are to be linked with already existing files. He may wish to extend or change the data coverage by adding, changing or removing fields of the user defined area of his records: PLUTO File Reorganization attends to all these changes. Records are re-formatted and such information as is still relevant in the new format is retained on the file. It is also desirable to reorganize any direct access file from time to time in the interests of efficiency. Under PLUTO, 'erased' records are retained on the file until reorganization, in order to save a great deal of link-changing at the time of erasure. Sets in PLUTO indexes may grow more than anticipated, so that continuation buckets are opened and time for access by unit name is increased. Some structure chains will gradually spread across the file. For these reasons it becomes desirable in time to reorganize PLUTO files for greater efficiency.

FACILITIES OFFERED BY PLUTO FILE REORGANIZATION

Individual file reorganization

Although reorganization of a PLUTO file involves changes to other files, the File Reorganization program allows the user to reorganize files individually. Thus space requirements are minimized and inquiries are still possible.

Connection and disconnection of files

PLUTO File Reorganization enables the user to connect a new master or structure file to his system, or to disconnect a master or structure file from his system. Since all records on a master file have a link area for each associated structure file, to connect or disconnect a structure file involves changes to master file unit record formats, so that reorganization of the associated master file or files is necessary. To connect or disconnect a master file is simpler, as it need not invlove changes to structure records.

Alterations to record formats

The user may reorganize a file in order to change the format of the user defined area, by adding or deleting testwords, condition codes, or unit or structure information. PLUTO File Reorganization retains as much of the existing information as is relevant in the new format. Other fields will be set to initial values.

PLUTO records have numerous links, not only within the file, but also to other files. When one file is reorganized, links to that file in other files which reference it must also be changed. Consequently, reorganization of PLUTO files is a relatively complex process. The decision to reorganize PLUTO files in the interests of efficiency is a matter of judgement; no firm rules can be laid down. Reorganization is simpler if carried out when files have not degenerated to too great an extent, but against this must be set the cost of reorganization in machine time.

GENERAL PRINCIPLES

The PLUTO File Reorganization programs do not destroy the old file which is always available to any program. Naturally any additions or changes to the file which take place while reorganization is in progress must be carefully monitored, and may have to be written again to the new file once reorganization is complete.

The new file can be written to magnetic tape in the first instance. Other measures have been taken to minimize the on-line storage requirement because this part of PLUTO may make the biggest demand for on-line storage. The file being reorganized must, obviously, be on-line, together with a number of short work-files and sorting space for the largest of these work-files. Some of the sort segments (e.g. link-resetting) can be run in parallel if multi-programming is in use.

Index

Activity chains 49ff.,54-55,60
Alterations to record formats 29,63
Applications 13,15-19

Basic functions 59ff.
Basic quantity 8,11,60,61
Basic record handling 59
Basic retrieval 23,59
Basic structure exploration 60
Bill of Materials 2,3-4,5,12-13
Bucket indexing 25
Bucket length 9
Buckets, Continuation 27
Bulk purchasing 18

Capital equipment manufacturing 15
Chain functions 60
Changes in design 3-5
Changing data 29-31
Character or Word fields 21
Codes: see Condition codes, Low level codes
Commitments 15
Comparison conditions 17,21,40
Component, definition of 1
Compound conditions 17,40
Condition codes 9,21,40,60,61
Conditional structures 17
Conditions 9,17,21,40,59-60,61
Connection and Disconnection of files 27,63
Continuation buckets 27,63
Contracts file 15
Conventional Direct-Access systems 5
Conventional data-processing methods 3
Costing 17
Counter 51,52,53,55,57,59

Design Changes 3-5
Direct-Access files 5,9,13,19,25ff.,41ff.,63
Direct components 3,10,11
Direct used-ons 3
Disconnection of files 63
Discounts 18
Duplication of records (conventional systems) 3
Duplication, avoided by PLUTO 4

Exchangeable Disc Store (E.D.S.) iii,19,27,41,45,46,47
Electrical installations, manufacturing and sales 17
'Erased' records 31,63
Examples 11,15-19,45,47-49,51-58,60-61
Explosion 5-7,13,17,18-19,23,34-38,42,45-47 49-51,60-61

Fixed Disc Store (F.D.S.) iii,19,27,41,45,47
File connection 9,27,63
File creation 27
File header 25,26,27,31
File maintenance 5,25-40
File modification 27-31
File reorganization 29,31,32,63-64
Flexibility 13
Formats of data-fields 20-23

Group data 23,27
Group dependent unit information 21,23
Group headers 25-27,29,31,59
Group index 25-27,29
Grouped records 9,21-23,59

"High-level" retrieval: see Standard retrieval

Implosion 5-7,13,18-19,23,39,43,47,49,51
Indented Explosion 5-6,45-47,59
Indented Implosion 5-7,45,48-49
Indexes, PLUTO 25-27
Indirect components 3
Indirect used-ons 3
Input 4
Integrated Information Systems 3,15
Item (Purchasing) 18-19

Labour costs 17
Layout strings 21
Level in structure: see Low level codes
Level of indenture in indented retrieval: see Relative level
Levels of retrieval: see Basic retrieval, Standard retrieval
Links 1,9-11,31-40,63
Loading 27-29
Low level chains 29,60
Low level codes 29,49,53,55-8,60
"Low Level" retrieval: see Basic retrieval

Maintenance 5,25-40
Master file 9 and passim
Master file unit record: see Unit record
Master record: see Unit record
Master record links: see PLUTO links
Material costs 17
Material requirements 3-4
Modifications in product design 3
Modifier 21,40

Named fields 21,22,59
Neighbours 11,30-39
Numeric data 22

On-Demand enquiries 4
Operations 12-13,15-17,22
Orders 12-13,18-19
Orders in Progress 12-13
Output (character peripherals) iii,4,18-19
Output (magnetic media) iii,18-19,41,42-58
Outstanding orders 12-13

Parameters presented to retrieval functions 41
Parent 3,11,30,32,34,36
Parent master file 13,17
Parent structure chain 30-36
Parts, parts listing 1,3,12-17,22
Parts and Stock file 12-13
PLUTO area 8-9,30,52,53,55,59
PLUTO basic functions 5,23,51-61
PLUTO fields 21
PLUTO files 9-11 and passim
PLUTO links 11,25,31-40,60,63
PLUTO records 9-10 and passim
PLUTO standard retrieval 5-7,41-58
Product numbers 17
Product structure information 3-4,5-7,13,15-17
Production control 12-17
Purchasing 18-19

Quantity, quantities 2-4,5-7,8,11,18,41ff.,52,63
Quantity modifier 21,40
Questionnaire 17

Record header 10,49-61
Record length 9
Record number 60
Relative level 6,46-49
Repeatable fields 21
Requirements 3-4,12-15
Retrieval 5-7,23,41-61
Routings 1,5,15-17,22,61
Run numbers 51ff.,59

Security keys 21,41
Sequence, sequence number 5,8,11,13,16,22,61
Single Level Explosion 5,42ff.,60-61
Single Level Implosion 5,42ff.
Sources of supply 1,18-19
Standard explosions 44-58
Standard retrieval 41-58
Stock 12-13,15
Stock File 12-13,15,22
Structure chain 11,15-17,32,37
Structure data 1,20
see also User's structure information
Structure file 9 and passim
Structure links 1,17
Structure record 9 and passim
Structure record header 8,9
Structure record links 25,30-40
Structures 1,6,10 and passim

Subunit 3,11,30,32,34ff.
Summarised Explosion 7,49-58
Summarised Implosion 7,51
Supervisory index 25-27
Supplier 3,12,18-19
Symbolic Names 21,41

Tags 41-58,61
Terminology 1-3
Testwords 8-11,20-23,40,41,59,61
Tools 14-17
Total Indented Explosion 5-6,45-47,59
Total Indented Implosion 5-7,45,48-49
Total Summarised Explosion 7,49-58
Total Summarised Implosion 7,51
Transport costs 18-19

Unit 1 and passim
Unit information 8-9,20-22,59,61
Unit Name 8-9,25,41,42,45,49,59-61
Unit record 9 and passim
Unit record header 8,9,52,53,55
Up-to-date information 4
Use of Output File 42
Used-on, definition of 1
Used-on block 2-3
User defined area 9-11,20-23
User's structure information 20-22

Variable-length fields 9,11
Variable-length records 9,11,13,15
Variants 5,17,59,61
Vendor 12-13,18-19
Vendor rating 18

Where-used, where-component 3
Work centres 12-17
Workfile 19,23,41ff.
Work in Progress 14-15
World market 17

AMENDMENT RECORD

List Number *Item Numbers*

	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to
	to

Should any additions or amendments to this Manual be necessary, they will be issued in the form of Amendment Lists. When such changes have been incorporated, an appropriate entry should be made in the table on the left and the pink instruction sheet(s) should be filed behind this page.