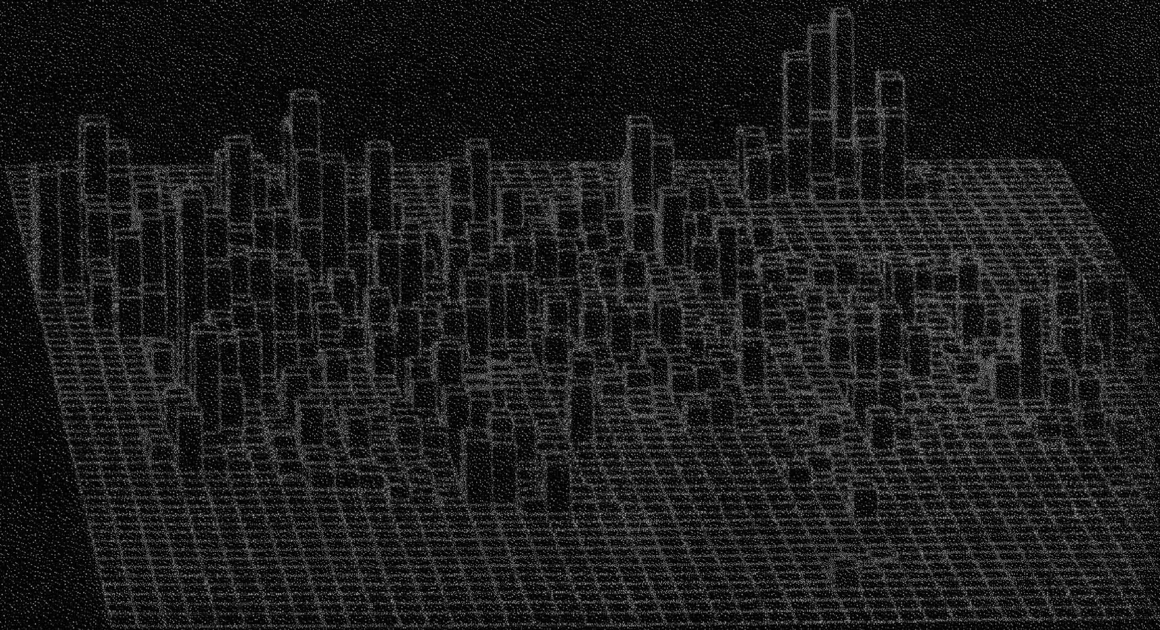
CRU

Data interchange on industry compatible tapes

by M. Visvalingam, M. J. Norman, and R. Sheahan



Census Research Unit
Department of Geography
University of Durham

Working [illegible]

The Census Research Unit, Department of Geography, University of Durham, is a small group of research workers investigating aspects of the theory and use of census data. It is currently funded as a research project by the Social Science Research Council.

The diagram on the cover represents total population per 1 km grid square in the northern part of County Durham: the height of each column is proportional to the population in that square. The county is viewed from the west, Gateshead being at the extreme left margin, West Hartlepool at the far right and Bishop Auckland at the centre-right. The original surface was calculated and drawn by computer.

UNIVERSITY OF DURHAM
DEPARTMENT OF GEOGRAPHY
CENSUS RESEARCH UNIT

WORKING PAPER No.6

MARCH 1976

DATA INTERCHANGE
ON INDUSTRY-COMPATIBLE TAPES

M. VISVALINGAM
(Census Research Unit)

M.J. NORMAN
(Computer Centre, University of Hull)

R. SHEEHAN
(Computer Unit, University of Durham)

Not to be quoted without the authors' permission

CONTENTS

DATA INTERCHANGE ON INDUSTRY-COMPATIBLE TAPES

1. INTRODUCTION

With the computer assuming a progressively more important role in data handling from the storage and management of data to its subsequent analysis and display, there has also been a corresponding increase in the flow of programs and data between computer systems of different manufacturers. The implementation of software packages such as SYMAP and SPSS over a wide range of machines and the centralised processing of data, such as census data, by public bodies, for subsequent dissemination in computer-readable form to diverse computing environments are but a few examples of the scale of data traffic between computer systems of different manufacturers.

The Census Research Unit (CRU) of the Geography Department, University of Durham is receiving the 1971 UK population census data on 'industry-compatible' (see below) tapes from the Office of Population Censuses and Surveys (OPCS). While OPCS processes the data on an ICL 1900 series computer, the CRU relies on the computing facilities provided by the Northumbrian Universities Multiple Access Computers (NUMAC), based on IBM 360 and 370 mainframe computers. The existence of standards for the physical properties of magnetic tapes has made the latter the most popular vehicle for the transport of large volumes of data. However, areas of incompatibility still remain as there are no similar specifications of standards for tape codes, label and block formats. This paper describes the problems of data interchange using the exchange of tapes between ICL 1900 and IBM installations and the conversion of the census tapes as specific examples.

2. PROBLEM AREAS

In our context, a reel of magnetic tape forms the physical unit of data transport between computer installations. Data are transferred from the core store of the supplier's computer onto magnetic tape using the conventions adopted by the software used in that system. The tape is then taken to the recipient's installation, where data

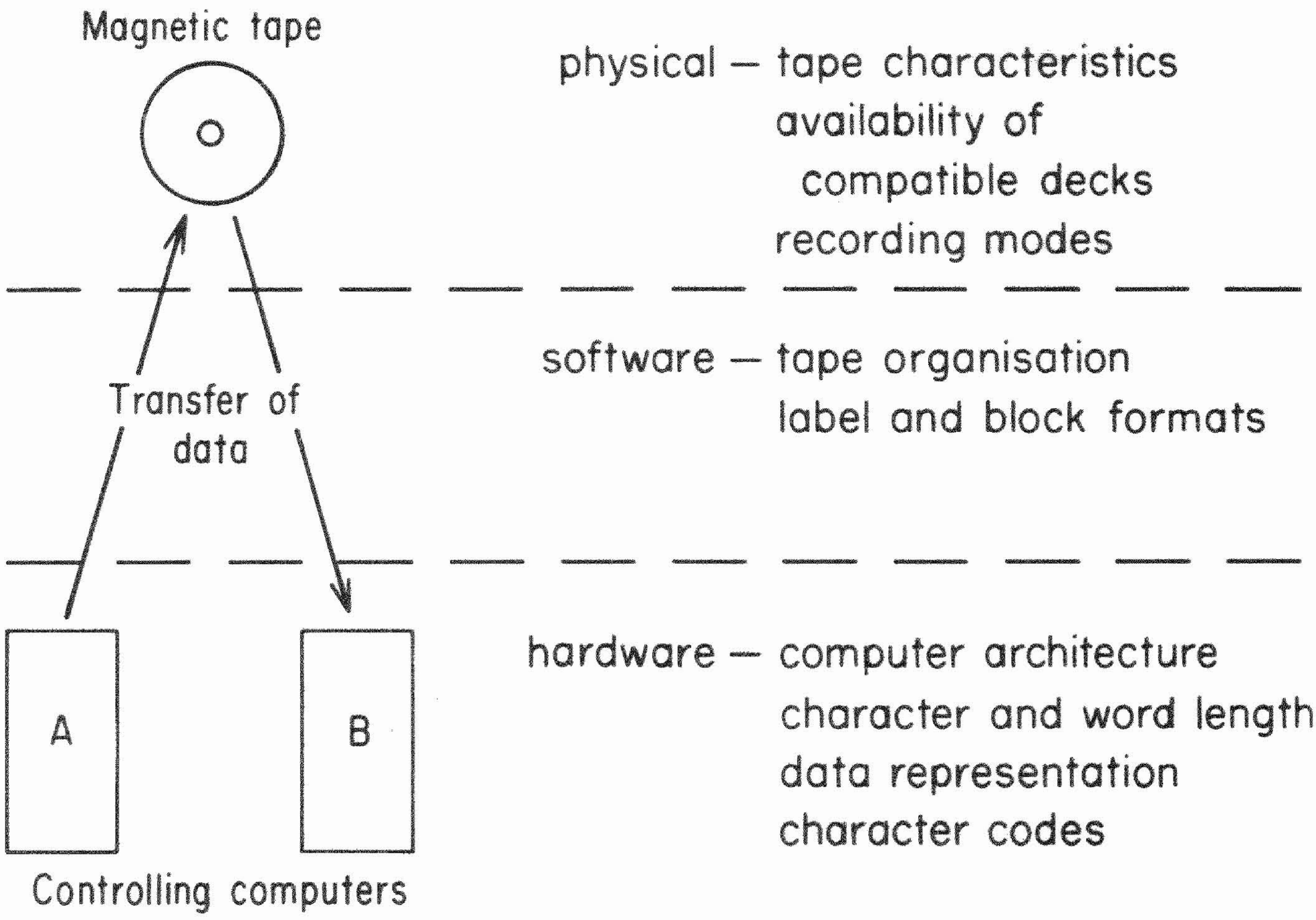


Figure 1. Problem areas in magnetic tape conversions

have then to be transferred from the magnetic tape into the main store of the recipient's computer using different software and conventions. In this process (Figure 1), three problem areas can be identified, associated with (1) the physical characteristics of the reel of magnetic tape, (2) the organisation of the core store of the controlling computer, and (3) the software conventions for processing the magnetic tapes.

2.1. Physical characteristics of magnetic tapes

The European Computer Manufacturers Association (ECMA), the British Standards Institute and the International Standards Organisation have defined the following standards (abstracted from ICL TP 4397[1]) for seven and nine track tapes :

	ECMA	BS	ISO DR
7 track	5	3968	1861
9 track NRZ1	12	4503/1	1863
9 track PE	36	4503/2	proposed

These specify the physical properties of tapes, such as spool dimensions, tape width and thickness, recording mode and density and positions of the reflective strips for the beginning- and end-of-tape marks. The relevant details can be found in the manufacturers' technical documents [1,2] Some of these are included in the proposed standard summary form for the description of magnetic tape files, described by King and Krasny[3]. The recipient of a 'foreign' magnetic tape would require the following details about the magnetic tape and the system used for recording:

2.1.1. Number of tracks

Tapes usually have either seven or nine tracks.

2.1.2. Recording mode

Seven-track tapes are recorded in non-return-to-zero-inverted (NRZI) mode. Nine-track tapes may be recorded in NRZI or phase-encoded (PE) mode (see ICL TP 4397, Chap. 1).

2.1.3. Parity

Regardless of the recording mode, all nine-track tapes are recorded with odd parity. Seven-track tapes may be recorded with odd or even parity (see ICL TP 4397, Chap.2 and IBM TP C28-6680-1, p.15). When using the ICL seven-track system with odd parity, an integral number of words must be transferred. Even parity is thus often used to transfer textual data, using the special character, $(octal 74), to terminate transfer. Binary data are normally transferred with odd parity because the chance occurrence of a bit pattern representing the above character would prematurely terminate transfer.

2.1.4. Recording density

Seven-track tapes may be recorded with 200, 556 or 800 bits per inch (bpi). However, with nine-track systems, NRZI tapes have a recording density of 800 bpi, while phase-encoded tapes have 1600 bpi.

2.1.5. Inter-block gaps (IBG)

Inter-block gaps are areas of uniformly magnetised tape which separate data blocks. Nine-track tapes have IBG of 0.6 inches. Seven-track tapes may either have short (0.56 inch) or long (0.75 inch) gaps. Inter-block gaps of 0.56 inch should not be used for data interchange as some fast tape decks cannot read tapes with short gaps.

2.1.6. Data conversion / translation features on IBM seven-track systems

IBM seven-track tape drives on the System 360 have an optional data conversion mode [2,3] in which three 8-bit main storage characters are written to tape as four 6-bit characters. Data conversion is mutually exclusive with the data translation feature, which translates the 8-bit main storage character to 6-bit BCD characters. The data conversion feature must be used with variable length tape records as the length field contains binary data.

Installations receiving foreign tapes must also have magnetic tape decks that are capable of processing the tape. For example, until OPCS acquired the facilities for processing nine-track tapes, the CRU had to copy the seven-track tapes to nine-track ones at a third installation with both systems, as NUMAC could only cope with nine-track tapes. Currently CRU receives census data on nine-track (PE) tapes. Hence data are recorded with odd parity at 1600 bpi with inter-block gaps of 0.6 inch.

2.2. Core Store Organisation

Data are written onto magnetic tapes via the core store of a controlling computer. While in concept data types are basically similar, the actual representation within the computer varies. This is in part an outcome of the differences in the architecture of

TABLE 1 : DATA REPRESENTATION (adapted from ICL Technical Publication No. 4105)

Type	ICL 1900s	IBM 360 (and 370)
Characters :	6 bits	8 bits (1 byte)
Integers:		
normal mode	48 bits reserved 24 bits used	32 bits (fullword)
other modes	24 bits reserved (compress integer mode)	16 bits (halfword)
Floating point numbers:	48 bits	32 bits
exponent	9 bits	7 bits
base of exponent	2	16
excess factor	256	64
argument length		
normal mode	38 bits	25 bits
double precision	69 bits	57 bits
Logical variables :	1 bit used	1 bit used
normal	48 bits reserved	32 bits reserved
other	24 bits (compress logical mode)	8 bits (1 byte)

TABLE 2 : INTERNAL MACHINE CODE COMPARISON

(adapted from ICL Technical Publication No.4397)

Character	360 code decimal	1900 code decimal
∇ space	64	16
. period	75	30
< less tha n	76	12
(left bracket	77	24
+ plus	78	27
& ampersand	80	22
! exclamation	90	17
$ dollar	91	60
* asterisk	92	26
) right bracket	93	25
; semi-colon	94	11
- hyphen/minus	96	29
/ solidus	97	31
, comma	107	28
% per cent	108	21
> greater than	110	14
? question	111	15
: colon	122	10
#	123	19
@ at	124	32
' quote/apostrophe	125	23
= equals	126	13
" quotes	127	18
A alphabetic	193	33
to		
I	201	41
J	209	42
to		
R	217	50
S	226	51
to		
Z	233	58
0 numeric	240	0
to		
9	249	9

machines. The IBM 360 and 370 computers are byte-orientated, where four 8-bit bytes form a 32-bit word. The ICL 1900 computer, is, on the other hand, organised primarily in terms of words, each word consisting of 24 bits (or four 6-bit characters). The different units and formats used for the storage of data elements in the ICL 1900 and IBM 360 (and 370) machines are given in Table 1. In addition,the definitions of the binary codes for representing graphic symbols are not consistent. The ICT 64-character set used by the ICL 1900s is defined for a character length of 6 bits, whilst an IBM byte provides for 256 EBCDIC symbols. Some of the graphic symbols (such as [,] , ↑ , ← in the ICT character set and several EBCDIC symbols such as \ , ¬ , ¢ etc) have no counterparts in the other system. In the past, the conversion of character codes from IBM BCD to ICT characters and vice versa was relatively easier as both provided for 6-bit codes. With ICL introducing the 2900 series computers capable of representing 8-bit ICL EBCDIC character codes, the translation of codes would again be somewhat easier. Wexler[4,5] discusses some of the problems in his proposal for character codes for use on the Scottish Regional Computing Organisation's 2980. So long as the character set used is a subset of the character sets available on different machines, the translation of character codes is a fairly straightforward task, given a copy of the graphic set used and the corresponding bit codes.

Thus, data items written onto magnetic tape reflect the architecture of the controlling computer and need to be re-organised for use at a foreign installation. As an example, the census data from OPCS consists largely of 24-bit binary integers (i.e. compressed integer mode) with some textual (character) data. The ICL binary integers were right-justified within IBM 32-bit full-word integers. This procedure checked the sign of the ICL number and set the leading byte of the IBM word to hexadecimal 'ØØ' or 'FF' for positive and negative numbers respectively (see Figure 2a). When the maximum value in a data set was not in excess of 32767, the data items were stored (truncated into) IBM halfword (16-bit) integers[6].

Words that contained character data were re-organised as shown in Figure 2b, using the internal codes for comparable characters listed in Table 2. The algorithm employed for indexing the corresponding character

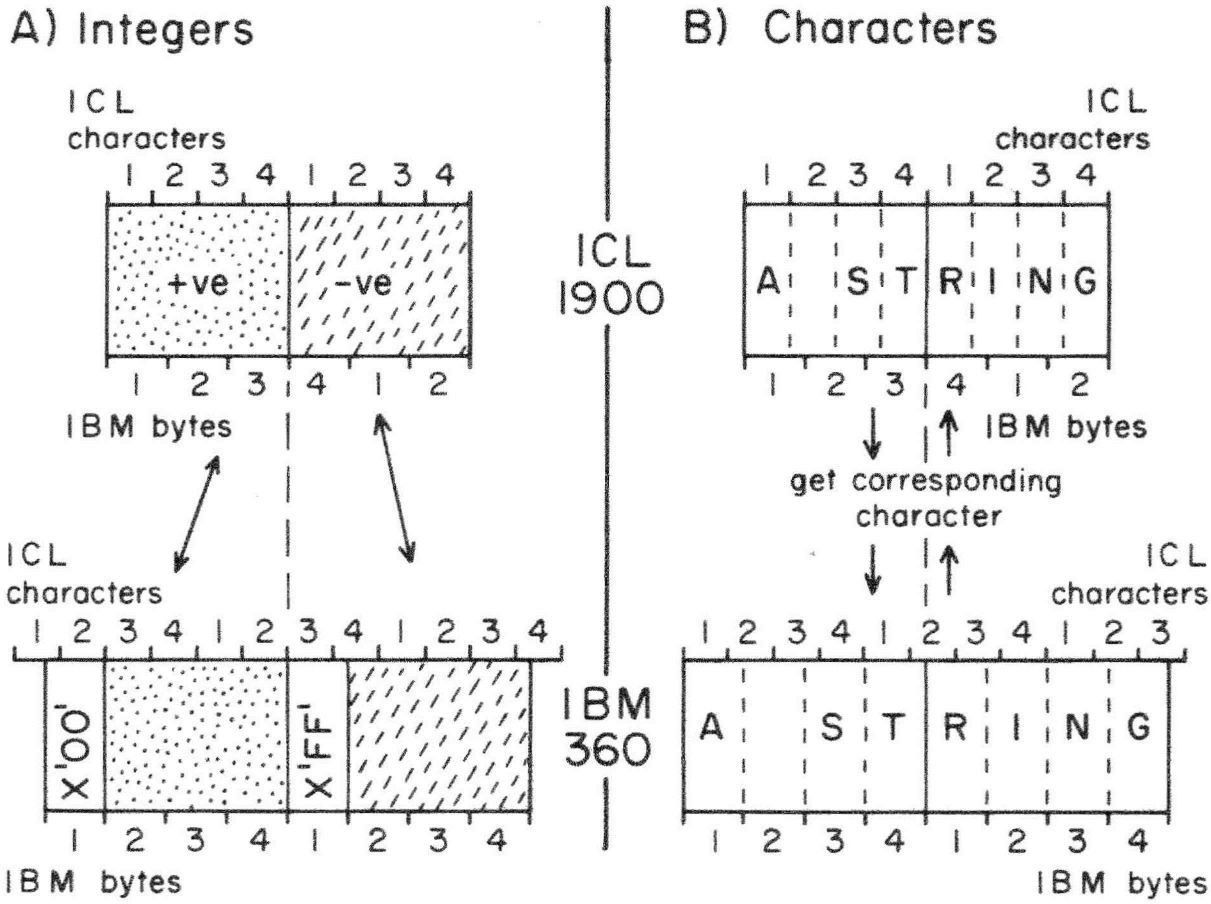


Figure 2. Reorganisation of ICL 1900 data items within IBM 360 core store

would depend on whether it were an ICL to IBM conversion or the reverse process and on the installation at which the conversion was effected. Users of *seven-track* tapes should note the ICL 1900 series hardware inverts the four most significant bits of a character during transfer. The Phase-encoded (PE) Population Census tapes from OPCS contained fixed length records with a mixture of character codes and binary integers and these were converted by a specially tailored FORTRAN program. Another, more general routine allows the user to define the format of the record and copes with ICL blocked records (contact M.V.). An ICL PLAN routine for converting IBM EBCDIC to ICL 1900 code is available at Hull (contact M.J.N.).

While binary integers and character codes are relatively easy to translate, the conversion of internal floating point representations is more complex. Not only do the mantissa and exponent vary in length but their relative positions within the allocated space also varies. Although in both systems the exponent is represented as an unsigned integer with sign given by use of excess factors, the base of the exponent varies. To complicate matters, both systems represent and normalise (see below) binary fractions somewhat differently, the difference in negative fractions being especially marked. There are further complexities associated with the loss of precision when converting from ICL 1900 48-bit floating point numbers to IBM 360 32-bit floating point representation, owing to the hexadecimal base of the IBM exponent (Hunter, 1975[7]). The following example is restricted to the conversion of normal or single length ICL floating point numbers (48-bits) to IBM 360 double length (64-bit) form (Figure 3).

Let us consider the internal representation of floating point numbers in the ICL 1900 computer. The number may be considered as consisting of two parts - a signed binary fraction plus an unsigned binary exponent - and occupies 48 bits (two contiguous words), the layout of which is shown in Figure 3. The leftmost bit is used for the sign of the number (Ø being + ve); the next twenty-three bits are the most significant part of the number; the next bit need not concern us here; the next fourteen bits are the least significant part of the number. Finally, the last nine bits are used for the exponent.

The binary point is assumed to be immediately to the right of the sign bit of the fraction. Negative fractions are stored in "2's complement" form, (the 2's complement is derived by changing each binary

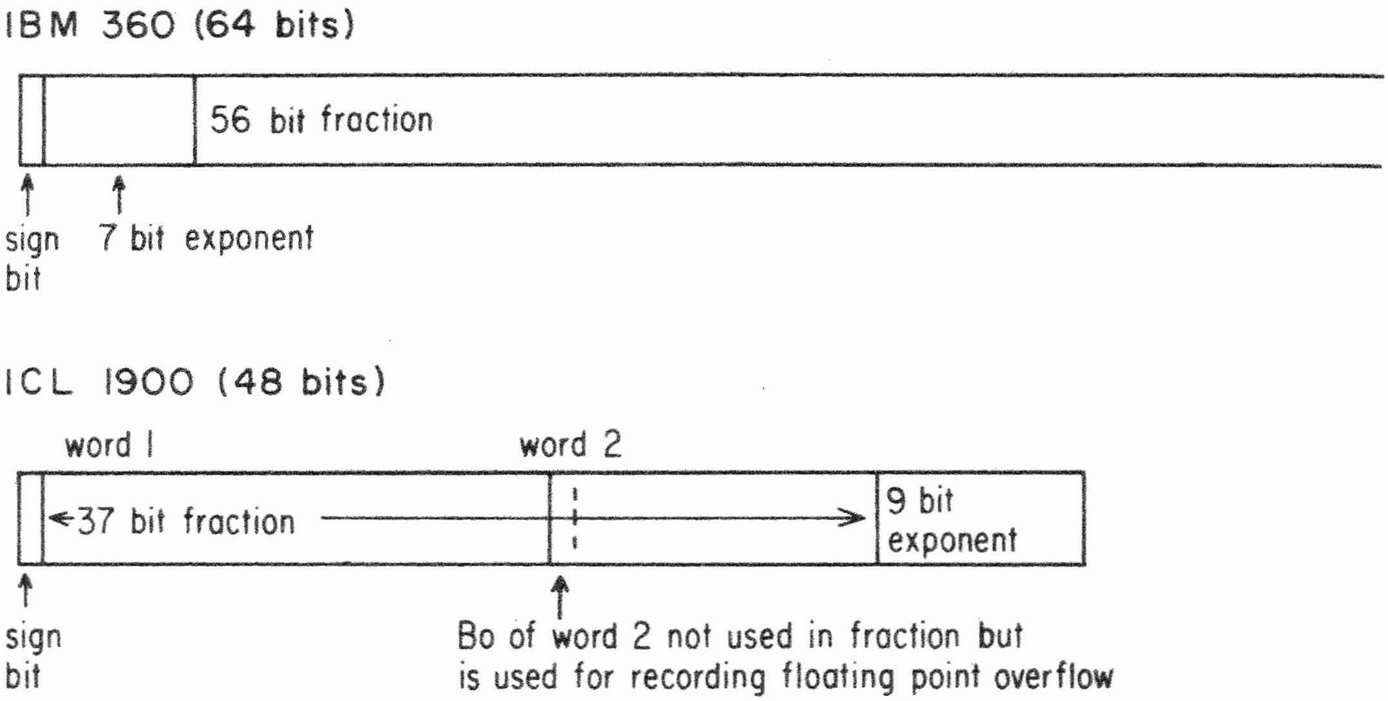


Figure 3. Internal representation of floating point numbers

1 to Ø and Ø to 1 in the original fraction and then adding 1 on the least significant position, e.g. 0.75≡011, -0.75≡101). The nine-bit exponent can represent values from 0 to 511; however, in order to allow for negative values, 256 is taken to represent Ø. Thus, in general, if (a) is the true signed exponent, it is represented by 256+a. This is known as the "excess 256 notation". The leftmost bit of the exponent (2^8) is the excess bit. Thus, the range of the exponent is from 2^{-256} to 2^{255}. The precision of the fraction is about 12 decimal places; the numbers are normalised, i.e. the leading bit is always 1 for positive fractions and Ø for negative fractions (the exponent being adjusted where necessary).

In 360/370 representation, short and long floating point numbers occupy 32 and 64 bits respectively. As with the ICl representation, the number is composed of two parts - a signed fraction and an unsigned exponent. However, IBM use a hexadecimal representation, i.e. the fraction is considered to consist of hexadecimal digits (Ø to F), each using four bits, and the exponent is interpreted as the power of 16 rather than of 2. As with the ICL representation, the sign of the number is indicated by the leftmost bit (Ø for + ve). However, the hexadecimal exponent follows immediately, occupying the next seven bits and the hexadecimal fraction occupies the remaining 56 bits; the hexadecimal point of the fraction is assumed to follow immediately after the exponent. Negative fractions are stored in the same way as positive ones, the difference being indicated by the sign bit. The exponent (b) is stored in excess 64 (i.e. b is stored as 64+b) notation and, as before, the excess bit is the leftmost bit (2^6) of the exponent. Thus, the range of the exponent is from 16^{-64} to 16^{63}. The precision of the fraction is about 16 decimal places. The number is normalized, i.e. hexadecimal shifts of the fraction ensure that the leading hexadecimal digit is always non-zero, both for positive and negative numbers.

It will be seen that, in absolute terms, the 1900 representation can hold slightly larger values than the 360/370 but the latter can hold slightly smaller values. Since these values are of the order 10^{76} and 10^{-76}respectively, problems associated with preserving magnitude are seldom encountered in practice.

The first step in the conversion is to examine the ICL binary exponent. If its true value is greater than 252, then the number is too large to be held by the IBM machine. In this case, the number is replaced by the largest possible IBM floating point number and the process terminated. If the ICL exponent is not too large, it is converted

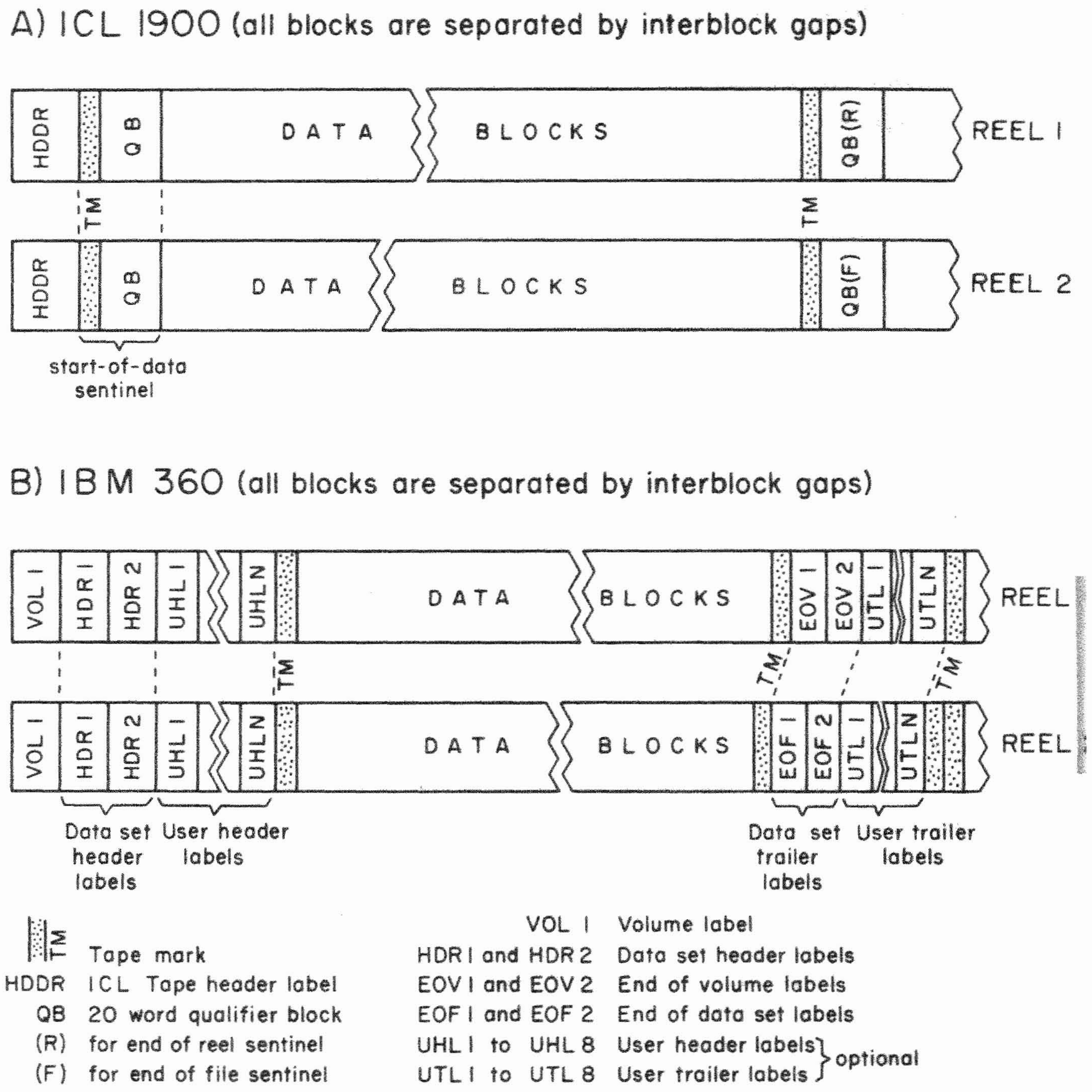

A) ICL 1900 (all blocks are separated by interblock gaps)
HDDR
TM
QB
DATA
BLOCKS
QB(R)
REEL 1
QB(F)
REEL 2
start-of-data sentinel
B) IBM 360 (all blocks are separated by interblock gaps)
VOL 1
HDR 1
HDR 2
UHL 1
UHLN
TM
DATA
BLOCKS
EOV 1
EOV 2
UTL 1
UTLN
REEL
EOF 1
EOF 2
REEL
Data set header labels
User header labels
Data set trailer labels
User trailer labels
TM Tape mark
HDDR ICL Tape header label
QB 20 word qualifier block
(R) for end of reel sentinel
(F) for end of file sentinel
VOL 1 Volume label
HDR 1 and HDR 2 Data set header labels
EOV 1 and EOV 2 End of volume labels
EOF 1 and EOF 2 End of data set labels
UHL 1 to UHL 8 User header labels
UTL 1 to UTL 8 User trailer labels
optional

Figure 4. Standard label format for a single data set spanned over two reels

from binary to hexadecimal by simply dividing the last eight bits by 4; the excess bit being ignored. The resulting quotient (q) and remainder (r) are preserved. The ICL excess bit is tested and if set ON it is switched off and the IBM excess bit is turned ON. The thirty-seven bit ICL fraction is formed, discarding the overflow bit. The sign of the fraction is tested and, if negative, the 2's complement of the fraction is taken [redacted] and the IBM sign bit is turned ON. If the remainder (r), from the division of the ICL exponent by four, is zero, then the quotient (q) - which is the IBM exponent - is correct and no shifting is required. If the remainder is non-zero, then the quotient must be increased by one and the fraction shifted right 4-r places in order to achieve the correct hexadecimal IBM fraction and exponent. Finally, the IBM exponent and fraction are combined to form the IBM floating point number (contact R.S.)

2.3. Tape Organisation

Although the organisation of information and data on a magnetic tape is primarily a function of software, as yet no standards have been adopted for block and label formats. For example, the label formats adopted by the IBM and ICL magnetic tape housekeeping systems are different. Moreover, while in concept both IBM and ICL software allow comparable blocking formats, actual implementations differ. Hence, to unscramble the relevant data from a 'foreign' tape, the details pertaining to the logical organisation of information on the magnetic tape must be known. These are best discussed under the following headings :

2.3.1. Label formats

Information on the organisation, formats and content of standard labels and the manner in which these are processed by the ICL and IBM magnetic tape processing systems can be found in references 1 and 2 respectively. The basic standard tape layouts for a single file (data set) spanned over two reels (volumes) are shown in Figures 4a and 4b. Labels identify the tape, the owner and data sets on the tape and also contain other information on blocking formats, number of blocks etc. Standard labels on IBM tapes are 80-character blocks, the first four characters of which identify the label (Figure 4b). A tape with standard labels will contain a volume and data set labels; user labels are optional.

A simple ICL tape file organisation (Figure 4a) starts with a header label at least nine words long, identified by the character HDDR in the first word. A tape mark, followed by a 20-word qualifier block forms the start-of-data, end-of-file, end-of-reel and user sentinels, which are differentiated by the state of the first word in the qualifier block. (Reference 1 also gives the structure of composite files on magnetic tapes adopted by the ICL 1900 system). Both systems also have provisions for processing non-standard and unlabelled tapes. The tapes are mounted with label processing disabled and the programmer is then responsible for defining or recognising the formats and processing the tapes with his own software. It should be noted that, on an unlabelled IBM tape (Figure 5), a single tape mark is interpreted as the end of a data set. Hence, the ICL tape (reel 1) in Figure 4a would be interpreted as having two files. ICL users should note that the end of the logical information on a tape should be terminated by at least two consecutive tape marks, so that the tape does not run off the end of the reel while it is being processed by IBM utilities.

In general, OPCS tapes have the simple file organisation shown in Figure 4a. These were identified by the tape-serial-numbers (tsn) quoted in the spools and amounted as non-standard tapes. As the recipient's housekeeping system does not check the labels of 'foreign' tapes, the header record was converted each time a tape was requested to verify that the correct tape was mounted by checking the file name, generation number, reel sequence number and the character representation of the tsn (see Reference 1 for details). The tape was then positioned past the qualifier block of the start-of-data sentinel, ready for reading.

2.3.2. Block formats

A record, consisting of a group of data items, is the basic unit of information which is exchanged between a program and input/output routines. For a card reader or punch, a record is the card, containing 80 characters. The basic unit of transfer between the magnetic tape and main store is a block, which usually contains an integral number of records. However, it sometimes may not relate in any way to the record structure. Blocks are separated from each other by inter-block gaps. In an ICL unbatched (or IBM unblocked) file, each record is written out as a block. With a recording density of 1600 bpi, 80 character blocks would occupy (80/1600=) 0.05 inch, and would be separated by inter-block gaps of 0.6 inch. To reduce the number and hence the space occupied by inter-

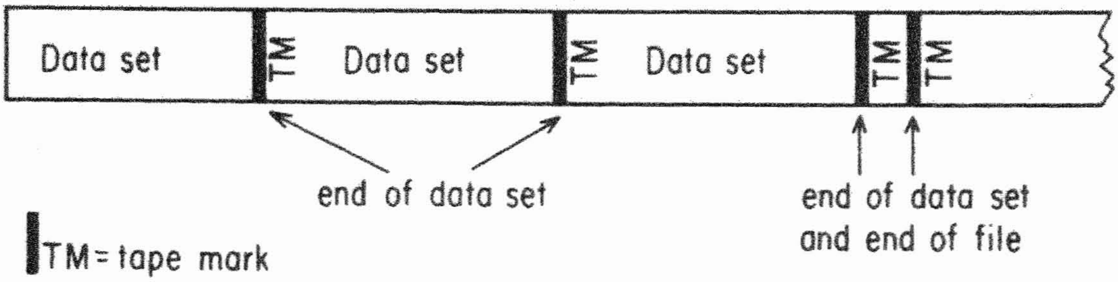


Figure 5. Multiple data sets on an IBM unlabelled tape

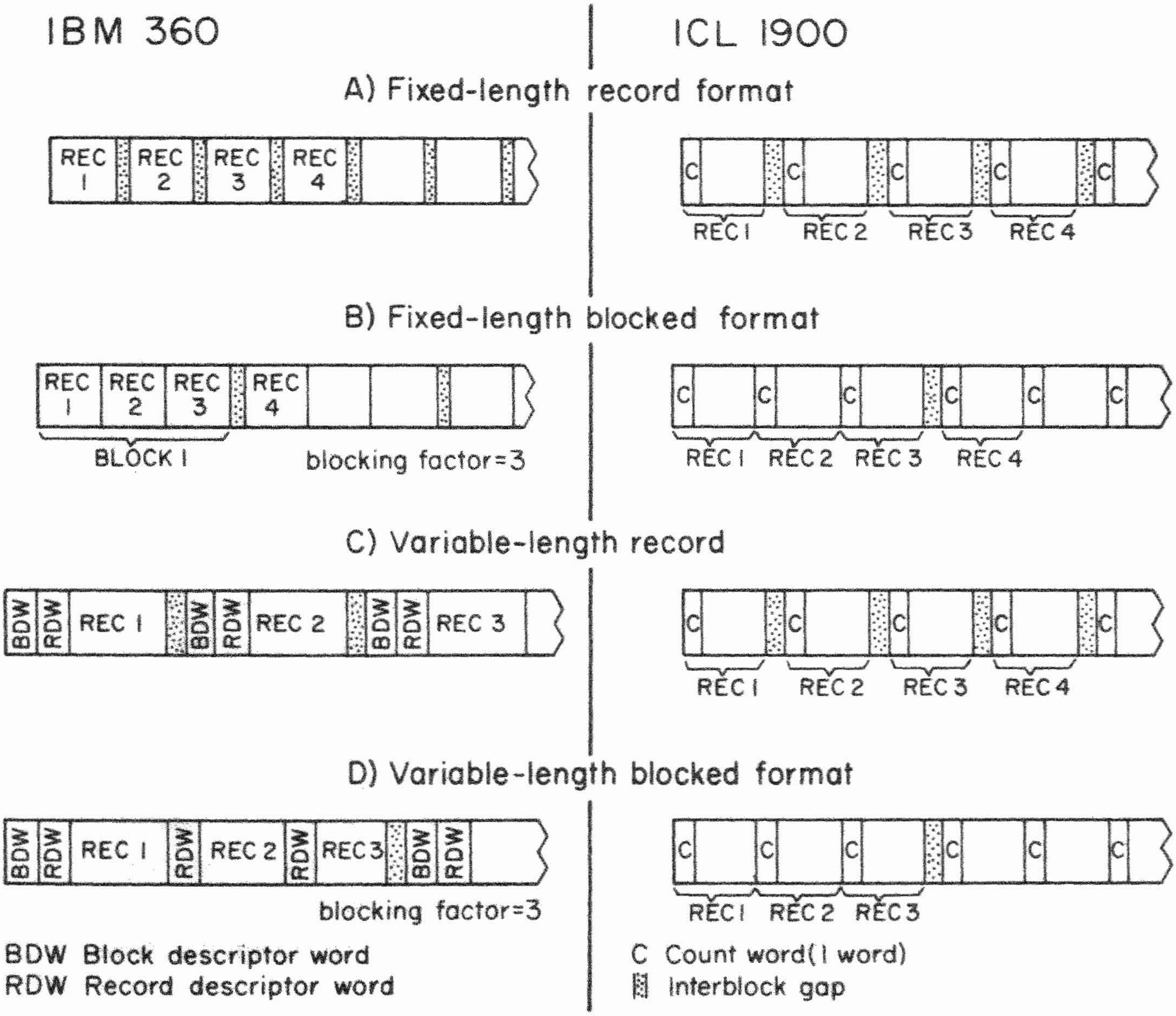


Figure 6. Blocking formats on tapes

block gaps, short records are usually combined into one block. This process of "blocking" also increases the efficiency of input/output operations by reducing both real and CPU time, as relatively fewer calls are made to the input/output controller. On the other hand, large blocks require a larger area in main store and increase the time taken to correct errors. Hence,while it is wasteful to write very short records, it is also inadvisable to write blocks which are too long. The size of the block and the blocking factor are determined by the user but hardware limits the minimum and maximum size of blocks. In the ICL 1900 system, the minimum length is 5 words (20 ICL characters or 15 IBM bytes) and the maximum size of a block is theoretically 32,767 words. However, software systems impose their own limits. In the IBM System 360 and 370, the limits are a minimum of 18 bytes and a maximum of 32767 bytes. Hence ICL users should note the narrower limits imposed by the IBM system. Also, when a computer reads (with odd parity) a block that does not fill an integral number of words, it may either truncate the block or fill the remainder of the last word with some padding character. Hence it is best to choose a block size which is likely to fill an integral number of words on most computers[8].

Moreover, while IBM and ICL tape housekeeping systems describe similar blocking concepts, it is evident from Figure 6 that the formats are quite different even for the basic fixed-length unbatched records. The formats diverge more markedly with blocking, especially when a logical record is spanned over several physical blocks (variable-length-spanned in IBM and multi-block record format in ICL). Furthermore, utility programs (such as *SORT in MTS and COPYOUT under George 3 and ICL's#XKYA) and high level programs such as PL/1 and FORTRAN use different blocking schemes as standard formats. It may be advisable sometimes to re-block the records to suit the home system.

The census tapes from OPCS contained unbatched variable-length records. Conceptually this corresponds to the variable-length record format adopted by IBM but, as is apparent from Figure 6, it does not correspond in implementation to any of the IBM block formats. Hence, these were read using the undefined formats, which transfers the whole block into the user's buffer area.

3. CONCLUSIONS

While industry-compatible tapes expedite the exchange of data between computer systems of different manufacturers, incompatibilities in data, block and label formats necessitate a 'conversion' process. Differences, which may require particular attention, have been picked out within the general scene of magnetic tape use, quoting existing discrepancies between the ICL 1900 and IBM 360 and 370 systems as specific examples. Users of other systems may be interested in the assorted experiences of King and Krasny[3], McLeod [9], and Macfarlane [10] with some other systems. Changes in the design of tapes and mainframe computers would undoubtedly affect data transferability on magnetic tapes. However, given the existing situation, any initiative towards standardising label and especially block formats would be very welcome. In the interim, the problems can be considerably alleviated by 'adequate' descriptions of the physical characteristics of the tape, its logical organisation and details of the representation of data types used (these can be quite easily abstracted from the manufacturer's manuals). In addition the name of a contact in the supplier's installation may prove highly valuable. King and Krasny[3] and McLeod[9] list most of the essential information.

ACKNOWLEDGEMENTS

The writers are grateful to Dr. D.W. Rhind of the CRU Durham, and Dr. R.S. Baxter of the Building Research Station, Watford, for their interest and comments on the paper. We would also like to thank Mrs. J. Dresser (CRU) who patiently typed the several drafts and Mr.A. Corner of the Geography Department, Durham, for preparing the diagrams for publication.

REFERENCES

1. ICL Technical Publication 4397 : Magnetic Tape, 1975 (1st edition TP4091, 1968)

2. IBM Technical Publication C28-6680-1 : IBM System/360 Operating System - Tape Labels, 1969

3. KING, H and KRASNY, M., 'A standard description for magnetic tape files', Annals of Economic and Social Measurement, Vol. 4, No. 3, pp. 449-454, 1975

4. WEXLER, J., 'Character codes', RCO 2900 Project document 26.2. 5, March 1975

5. WEXLER, J., 'A proposal for character codes for use on the Regional 2980', TAP/75/22, May 1975

6. VISVALINGAM, M., 'Storage of the 1971 UK Census data; some technical considerations, CRU Working Paper 4, Department of Geography, University of Durham

7. HUNTER. G., 'A quantitative measure of precision', Computer Journal, Vol. 18, No.3, p. 231-233, 1975

8. WAITE W.M., 'Hints on distributing portable software', Software - practice and experience, Vol.5, p. 295-308, 1975

9. McLEOD, R., 'Transferring data between institutions', IUCC Newsletter, Vol. 4, No.1, 1975, p.15-17

10. MACFARLANE, A., 'What is an industry-compatible magnetic tape ?, IUCC Newsletter, Vol. 4, No.1, 1975 p.18.