XXDP+ File Structure

PRODUCT NAME; AC-S866A-M0

PRODUCT NAME; CHQFSAO XXDP + FILE STRUCT DOC

PRODUCT DATE: APRIL 1981

MAINTAINER: DIAGNOSTIC SYSTEMS ENGINEERING

DOCUMENT: XXFORM. MEM
DATE: 17 FEB 1981
CHARGE NUMBER: P98-08823

XXDP+ FILE STRUCTURE SPECIFICATION
REVISION 0.0

MAINTAINED BY: DIAGNOSTIC SYSTEM ENGINEERING

REVISION HISTORY
REVISION 0.0 17-FEB-1981
by Michael Cinnamon

REVISION 1.0 8-OCT-1984
by Larry Pruche

TRANSCRIPTION HISTORY
REVISION 2.0 11-MAR-2021
by Ian Hammond (IJH)

Primary transcription source:

http://www.bitsavers.org/pdf/dec/pdp11/xxdp/AC-S866A-M0_CHQFSA0_XXDP+_File_Struct_Doc_Apr81.pdf

Additions and corrections source:

http://www.bitsavers.org/pdf/dec/pdp11/xxdp/diag_listings/AC-S866B-MC_CHQFSB0_XXDP+_File_Struct_Doc_Oct84.pdf

Contents

1.0 INTRODUCTION

The structure that XXDP+ uses for storing files on media is unique to XXDP+. The structure was originally based on DOS-11 but it has since been modified to accommodate the needs of XXDP+, although many similarities still exist.

XXDP+ supports both random access and sequential access type devices, A directory index structure is used for accessing files on random access type devices such as disks. For sequential devices like magtape a header record containing file information precedes each file.

2.0 DATA STRUCTURES

2.1 Data Blocks

The basic unit of data transferred in XXDP+ file I/O is a data block. A data block is defined as a group of 512(10) 8 bit bytes.

2.1.1 Random Access Devices

On random access devices, data blocks are addressed by their logical block number. Logical blocks are data blocks that are addressable in a linear ordered fashion. The order ranges from 0 to n-1 , where n is the maximum number of blocks supported on the device. The variable n varies depending on the device but is not greater than 65535 (10). This means that no matter how many data blocks physically reside on a device only the first n blocks are accessible.

Logical blocks are not necessarily the same as physical blocks. A physical block is usually limited to the amount of data a physical sector contains which may be less than 512 (10) bytes. Also a physical block houses at most one logical block.

A linked list of logical blocks is set up using linked blocks. A linked block is a logical block that devotes the first word of the block to contain a link word. The link word contains the logical block number of another block. A zero link word indicates that the block containing ft is the last block of the list.

2.1.2 Sequential Access Devices

On sequential access devices such as magtape and cassette, data blocks are stored as 512 byte records. Because XXDP+ uses the same read/write routines for both random and sequential access devices, logical linked blocks are stored on sequential type devices.

However, the link word is only some non-zero value to indicate that there are more blocks in the list. A zero link word indicates the last block of the list.

3.0 FILE STRUCTURES

3.1 Types Of Files

There are three types of files supported by XXDP+. They are contiguous files, text files and binary files. This is not to mean that other types of files cannot be stored on an XXDP+ medium, but XXDP+ only has the capability to produce these types.

3.1.1 Contiguous Files

A contiguous file is a set of logical blocks which physically reside immediately adjacent to one another on the media. The term 'immediately adjacent' is obvious for sequential devices but not quite as apparent for random access devices. For these it means for any logical block n, the next contiguous block is located at n+1 .

Contiguous files are normally used to store core image data such as the XXDP+ monitor.

3.1.2 Text Files

Text files are made up of a series of linked blocks. Each block contains 510(10) 8 bit ASCII characters. An ASCII null character (a byte with a value of zero) is used to indicate the end of the file.

3.1.3 Binary Files

Binary files are used to store executable programs. They are made up of a series of linked blocks each of which contain sections of the program. These sections are in absolute formatted binary and there is at least one per logical block.

The absolute formatted binary specification is as follows:

Byte 1	- Contains a value of 1 to indicate starting point.
Byte 2	- Contains a value of 0. This must follow byte 1.
Bytes 3 and 4	- Contains number of bytes (N) in this binary block. It includes bytes 1, 2, 3 and 4 but excludes the checksum byte.
Byte 5 and 6	- Contains the starting memory address where the following data bytes are to be stored.
Byte 7 to N	- Data bytes. N <= 509. The maximum number of data bytes is 503.
Byte N+1	- Contains the checksum byte. The checksum is the two's complement of the sum of the data in all N bytes. It is generated ignoring overflow and carry conditions.

The end of a binary file is indicated by a binary block that has a byte count of 6. Bytes 5 and 6 contain the program's transfer address, this block is known as the 'transfer block'.

A binary block with a byte count of 5 indicates a 'bias block’. Bits 0 and 1 of byte 5 represent bits 16 and 17, respectively, of the load address for the next binary block.

4.0 MEDIA STRUCTURE

4.1 Random Access Structure

All XXDP+ random access devices are set up to contain the following pieces of information: bootstrap, monitor core image, master file directories (MFD), user file directories (UFD), and bit maps.

The bootstrap is a program that always occupies logical block 0 on the device. It is a core image that is placed there by the utility command *$AVM'. The bootstrap knows where on the disk the monitor core Image resides.

The monitor core image is a contiguous file that is 16 blocks long, It is placed on the disk by the utility command * SAVM ' . Its position depends on the type of disk. The MFD is a table of information which contains pointers to the UFDs and the bit maps.

4.1.1 Master File Directory

The MFD is placed on the disk by the utility 'ZERO' command. The type of disk determines which of two varieties of MFD is used and where it is on the device.

MFD Variety #1

		Word offset into block
MFD1:	Link to MFD2	0
	Interleave factor	1
	Bitmap start block #	2
	Pointer to bitmap #1	3
	Pointer to bitmap #2	4
	…	…
	Pointer to bitmap #n	N+2
	0	N+3
	Unused	N+4 to 255

The interleave factor is used as part of the block allocation algorithm. Normally blocks are allocated contiguously if possible but on certain devices, the amount of time it takes to access a linked list of Mocks is reduced by placing each block of the list a constant number of blocks apart. This constant number is the interleave factor. The pointers in the table are the logical block numbers of the respective bit maps.

		Word offset into block
MFD2:	0	0
	401	1
	Pointer to first UFD block	2
	9 (10)	3
	0	4
	Unused	5-255

The first word is a link of zero indicating no more MFDs . The second word corresponds to the DOS-11 UIC [1,1]. The third is the logical block number of the first UFD block. The fourth is the number of words in each UFD entry; this is always nine (9),

MFD Variety #2

		Word offset into block
MFD 1/2:	0	0
	Pointer to first UFD block	1
	# of UFD blocks	2
	Pointer to first bit map block	3
	# of bit map blocks	4
	Pointer to MFD 1/2	5
	0	6
	Number of supported blocks	7
	# of blocks pre-allocated	8
	Interleave factor	9
	0	10
	Pointer to first block of monitor image	11
	0	12
	Track and sector address for bad sector file (single density)	13
	Cylinder address for bad sector file (single density)	14
	Track and sector address for bad sector file (double density)	15
	Cylinder address for bad sector file (double density)	16

The entry at offset eight means the number of blocks on the reserved for device structure information. The last four entries table refer to the location on the disk of the DEC STD 144 bad sector file.

4.1.2 User File Directory

The User File Directory (UFD) is a list of the files on the media. It is created by the utility 'ZERO' command. The UFD is arranged as a linked list of logical blocks and the number of blocks that the UFD occupies depends on the device. Each block of the UFD contains space for 28(10) file entries.

Link to next UFD block
File entry #1
File entry #2
…
File entry #28 (10)

Each entry is a table of 9(10) words that contains the following information about the file.

	Word
File name	1
	2
File extension	3
File date	4
ACT-11 logical end	5
First block	6
File length	7
Last block	8
ACT-11 logical 52	9

Words 1,2,3 - The file's six character name and three character extension encoded in rad50. A deleted file is indicated by a zero in these three words.

Word 4 - The DOS-11 format for the date given the file when it was put on the media.

Words 5 & 9 - ACT-11 use only. Not used in XXDP+.

Word 6 - The block number of the first logical block that the occupies.

Word 7 - The number of logical blocks that the file occupies.

Word 8 - The block number of the last logical block that the file occupies.

4.1.3 Bit Map

The bit map is a file that contains the current status of every supported logical block on the media.

The bitmap is arranged as a linked list of logical blocks. The number of blocks that the map occupies depends on the device. It is created by the utility 'ZERO' command. Only the first 64 words of each map block have meaning.

	Word
Link to next map bock	1
Map number	2
60(10)	3
Link to first map	4
Map for blocks 0-15 (10)	5
Map for blocks 16-31 (10)	6
…	…
Map for blocks 944-959 (10)	64

Word 1 - The logical block number of the next map block contains zero if it is the last map.

Word 2 - Which map this one is.

Word 3 - Number of words used for map.

Word 4 - The logical block number of first bit map.

Words 5-64 - Map for 960 blocks. Bit is set when block is used. Bit is cleared when block is free.

Word 65-255 - Not used.

4.1.4 Random Access Device Information

Device	Mnemonic	1st UFD block #	# of UFD blocks	1st bitmap block #	# of maps	MFD1	MFD2
TU58	DD	3	4	7	1	1	2
RP04,5	DB	3	170.	173.	50.	1	2
RK03,5	DK	3	16.	4795.	5	1	4794.
RL01,2	DL	24.	146	2	22.	1	-
RK06,7	DM	31.	96.	2	29.	1	-
RP02,3	DP	3	170.	173.	50.	1	2
RM03	DR	52.	170.	2	50.	1	-
RS03,4	DS	3	4	7	2	1	2
TU56	DT	102	2	104	1	100	101
RX01	DX	3	4	7	1	1	2
RX02	DY	3	16.	19	4	1	2
UDA50	DU	35.	234.	269.	69.	1	2
RDRX	DQ	3	16.	19.	4	1	2
RC25	DA	35.	181.	216.	53.	1	2

Device	# blocks on device	# blocks to pre-allocate	Interleave	Boot block #	Monitor block #
TU58	511.	40.	1	0	8.
RP04,5	48000.	255.	1	0	223.
RK03,5	4800.	69.	5	0	30.
RL01,2	20460 or 10200	200.	1	0	170
RK06,7	27104.	157.	1	0	127.
RP02,3	3000.	255.	1	0	223.
RM03	48000.	255.	1	0	222.
RS03,4	989.	41.	1	0	9.
TU56	576.	69.	5	0	30
RX01	494.	40.	1	0	8.
RX02	988.	55.	1	0	23.
UDA50	65535.	338.	1	0	23.
RDRX	790.	55.	0	0	23.
RC25	5084.	269.	1	0	3.

4.2 Sequential Access Devices

4.2.1 Magtape

Although magtape is not usually considered as a file structured device, certain structure features are implemented to enable creation and retrieval of multiple files.

The files on maqtape are terminated by an end-of-file mark (EOF) or tape mark (TM). The last file on the tape is terminated by two consecutive EOFs to indicate logical-end-of-tape.

1st file	EOF	2nd file	EOF	Last file	EOF	EOF

Each file on magtape is made up of a header and data records.

Header 7 words	IRG	Data record 256 words	IRG	Data record 256 words	IRG	EOF	IRG

The header record is structured as follows:

	Word
File name	1
	2
File extension	3
401	4
0	5
File date	6
File size	7

Words 1,2,3 - The file name and extension in rad50

Word 4 - DOS-11 UIC [1,1].

Word 5 - Set to zero

Word 6 - Date given the file when written on the tape. It is in DOS-11 format.
Bit 15 of the date, when set, indicates a contiguous file.

Word 7 - The number of logical blocks (records) in the file.

The first file on a magtape is normally the XXDP+ monitor core image. It is written as a contiguous file. Every new file is written at the logical end of tape. The last file on tape may be written so part ties after the physical EOT marker but no file will be written entirely after the physical EOT mark.

The number of files that a magtape can accommodate is a function of the file size, the drive density and the tape length.

4.2.2 Cassette (TU60)

Cassette is structured similarity to magtape in that each file is preceded by a header and a marker that identifies the logical end of tape. However, the actual data in the header and end-of-file marker are different.

The files on cassette are terminated by a tile gap. The last file on the tape is terminated by a sentinel file. A file gap muse precede the first file on the cassette. The tape is formatted as follows:

Gap	File A	Gap	File B	Gap	File C	Gap	Sentinel

Each file on a cassette is made up of a file header record and file data records in multiples of four.

File header 32 bytes	Data record 128 bytes	Data record 128 bytes	Data record 128 bytes	Data record 128 bytes	Gap

The file header record is structured as follows:

	Byte
File name	1-6
File extension	7,8,9
File type	10
100000 (8)	11,12
0	13-16
Date	17-18
File length	19-20
0	21-22

Bytes 1-9 - Filename and extension encoded in rad50.

Byte 10 - File type indicator. 0=6000, 14=Deleted.

Bytes 11,12 - Word indicating record length set to 100000(8) (128 byte records).

Byte 13 - Sequence number not used in XXDP+.

Byte 14 - Header continuation not used in XXDP+.

Bytes 15,16 - Not used.

Bytes 17-18 - DOS-11 formatted date given when put on the tape.

Bytes 21-32 - Not used.

5.0 GLOSSARY

IRG - Inter-record gap. The gap that is written between records on magtape.

MFD – Master File Directory

Rad50 - A method of encoding three ASCII characters into one 16 bit word.

UFD - User File Directory.

UIC - User Identification code.