APPENDIX F
AIM 65 AUDIO TAPE FORMAT

The AIM 65 audio cassette tape format is designed to provide fast, reliable recording and reading of both object code and source code. Object code is recorded in binary form, exactly as it appears in memory. Recording object code in binary, is twice as fast as recording it in ASCII, since one byte contains two hexadecimal numbers in binary format, whereas one byte contains only one hexadecimal number in ASCII format.

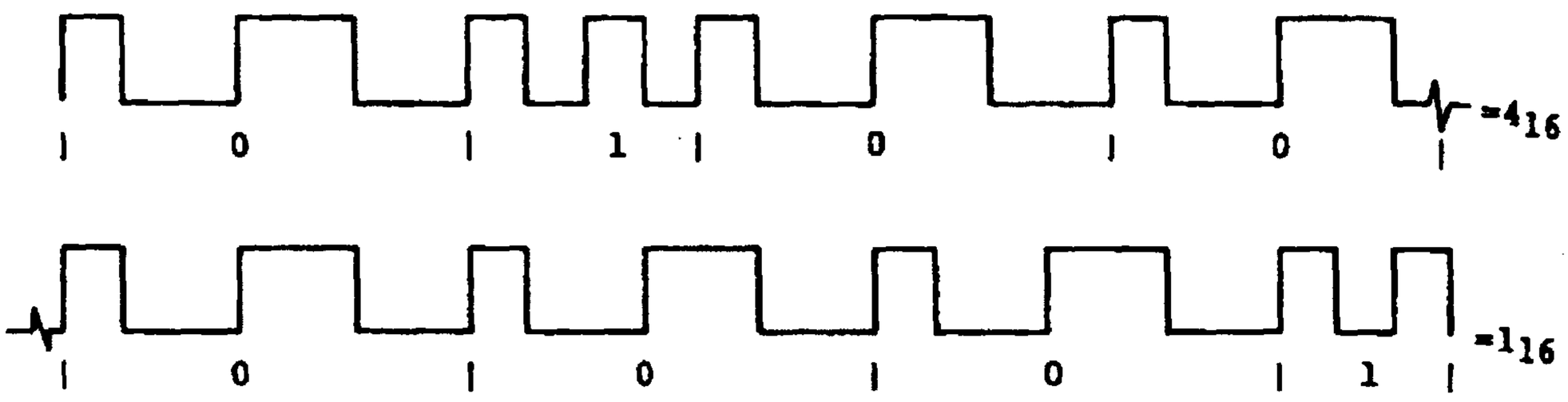
Source code is recorded in ASCII format, the form in which it appears in the Editor Text Buffer.

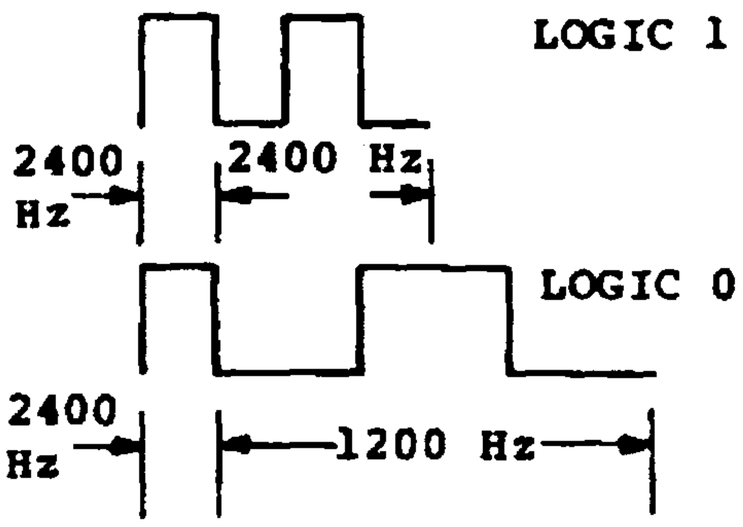
BIT LOGIC STATE DEFINITION

Each transmitted bit begins with a positive one-half cycle of 2400 Hz. tone. The following three half-cycles determine the logic state of the bit. Three half-cycles of 2400 Hz. equal a logic "1". Three half-cycles of 1200 Hz. tone equals a logic "0".

The following example shows eight bits of data. If this data represents one byte, the hexadecimal value of 41 equals two hexadecimal numbers, 4 and 1. If the data is in ASCII format, the character A is represented.

EXAMPLE:





F.1 BLOCKED FORMAT

The data is recorded in blocked format. One block contains 80 bytes of data and 36 bytes of synchronization, control, and block checksum information. The block format is:

SUBFIELD	SYN	#	BLK	OBJECT OR TEXT DATA	BLK CHKSUM
Value	1616...1616	23	HH	XXX...XXXX	HHHH
No. of bytes	(32)	(1)	(1)	(79)	(2)

SYN

The block begins with 32 bytes of Synchronous Idle (SYN) characters (ASCII 16). During read operations, the SYN pattern allows AIM 65 to sense the start of the block and synchronize to the incoming serial data stream.

SYN characters are also used to provide an interblock gap. The number of SYN characters is determined by the contents of address A409 (GAP). The default value of 08, established by a "cold" RESET, generates 32 SYN characters. This value provides a minimum gap for loading object code or source code into an empty Editor text buffer.

For assembling from tape or reading data into a partially-filled Editor text buffer,a larger gap size is required. This allows AIM 65 to stop the tape after a block has been read in order to process the data before reading another block. To lengthen the gap size additional SYN characters are required. The gap value in address A409 should be changed from 08 (32 SYN characters) to 80 (512 SYN characters). This number should be adequate for all audio cassette recorders, but a lower number may be suitable

for your recorder. That number can be determined by experimentation.

#

The character # (ASCII 23) denotes that the data on the tape is recorded in the AIM 65 format, as opposed to the KIM-1 format described in Appendix G.

BLK

The block count (BLK) defines the block number. This number starts with 00 on the first block and increments by one for each block recorded, in hexadecimal, to FF. If more than FF blocks are recorded, the number restarts at 00.

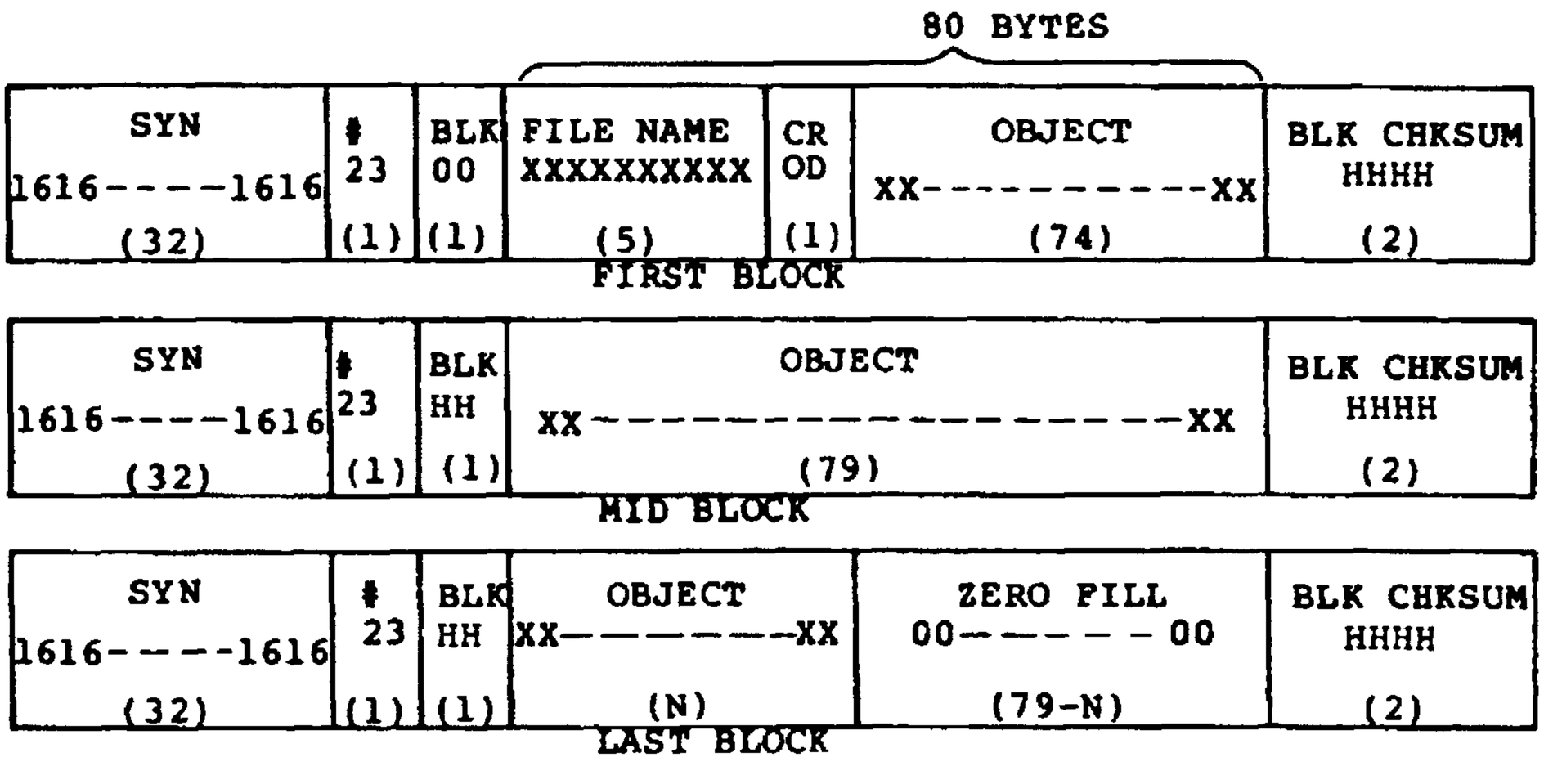
Data

The actual recorded data represents either source or object code. Within each data type are three unique data block types: First block, mid-blocks, and last block. See the object and text data record formats for detailed definition of the data format.

BLK CHKSUM

The block checksum (BLK CHKSUM) is the hexadecimal sum of the 80 data characters, truncated to four hexadecimal digits, (i.e., carry is ignored).

F.2 OBJECT DATA FILE FORMAT



FILE NAME

The file name (FILE NAME) consists of one to five ASCII characters that uniquely identify the file.

CR

The CR character (ASCII 0D) after the FILE NAME indicates that the data format is object rather than text.

Object Data Format

The object data consists of multiple object data records, each containing a starting address and up to 24 bytes of information. The object data is recorded in hexadecimal form. The object data includes both object instructions and data.

The object data record format is:

$$\text{Data Record: } ;N_1N_0A_3A_2A_1A_0\overbrace{D_1D_0}^{1}\overbrace{D_1D_0}^{2}\ldots\overbrace{D_1D_0}^{n}X_3X_2X_1X_0CR$$

$$\text{Last Record: } ;00C_3C_2C_1C_0X_3X_2X_1X_0CR$$

Where:

;	= Start of the record (ASCII 3B)
N_1N_0	= Number of data bytes in the record, in hexadecimal. The maximum number of bytes in one record is 18_{16} (24_{10}).
	= 00 for the last record.
$A_3A_2A_1A_0$	= Address of the first data byte in the record, in hexadecimal.
D_1D_0	= One 8-bit data byte = Two hexadecimal numbers.

$C_3C_2C_1C_0$ = Number of records in hexadecimal, including the data records and the last record.

$X_3X_2X_1X_0$ = Record checksum, in hexadecimal. This is the sum of all the characters in the object code record except the ; character and the record checksum. The checksum is truncated to four hexadecimal digits, i.e., carry is ignored.

CR = Carriage Return (ASCII 0D) which indicates end of record.

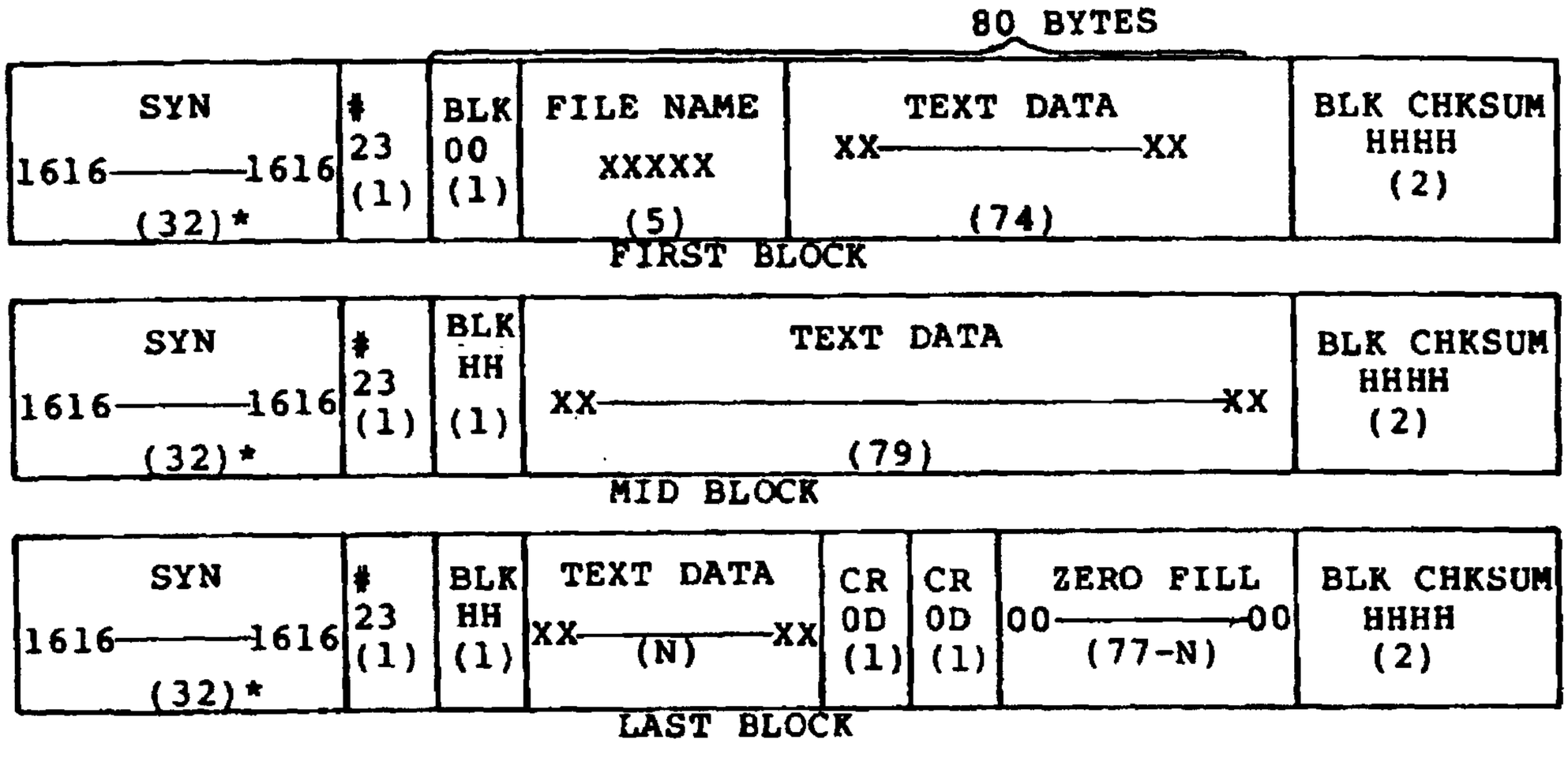
All files contain at least two object code records: the first record and the last record. The last record uniquely identifies the end of the file data.

Since each object code record contains a starting address, various portions of memory can be recorded in one file. Programs or program segments residing in different parts of memory may therefore be recorded on the same file. This simplifies subsequent memory loading procedures, as well as saving load setup time.

Zero Fill

After the last data record is recorded, the remaining data bytes are filled with hexadecimal zeros.

F.3 TEXT DATA FILE FORMAT



*The value shown corresponds to a gap size in $A409 (GAP) of $08.

FILE NAME

The file name (FILE NAME) consists of one to five ASCII characters that uniquely identify the file.

TEXT DATA

The text data consists of characters recorded from the Editor Text Buffer. The data is recorded in ASCII format as it exists in memory. The text data may be the source program for input into the assembler or any text information.

CR

The CR character (ASCII 0D) indicates end of a text record in the text buffer. CR will appear throughout the text buffer separated by no more than 60 characters. Two CR's in succession indicate end of the text file.

ZERO FILL

After the end-of-file indication, the remainder of the block is filled with hexadecimal zeros.