



FORMATTED BINARY PAPER TAPE

1. INTRODUCTION

This Programming Note describes a standard CA data representation called Formatted Binary as it appears on punched paper tape. Formatted Binary Paper Tape is supported by the Operating System and by the IOS sub-system of the Real Time Executive. It can also be handled by a CA FORTRAN program which directs unformatted READ and WRITE requests to a paper tape device.

Stand-alone programs written in Assembler Language can process Formatted Binary Paper Tape by using any technique which conforms to the specifications of this Programming Note.

2. FILE STRUCTURE

A complete file always has these components:

Leader

Data Records, each followed by an Inter-record Gap

End-of-File Record

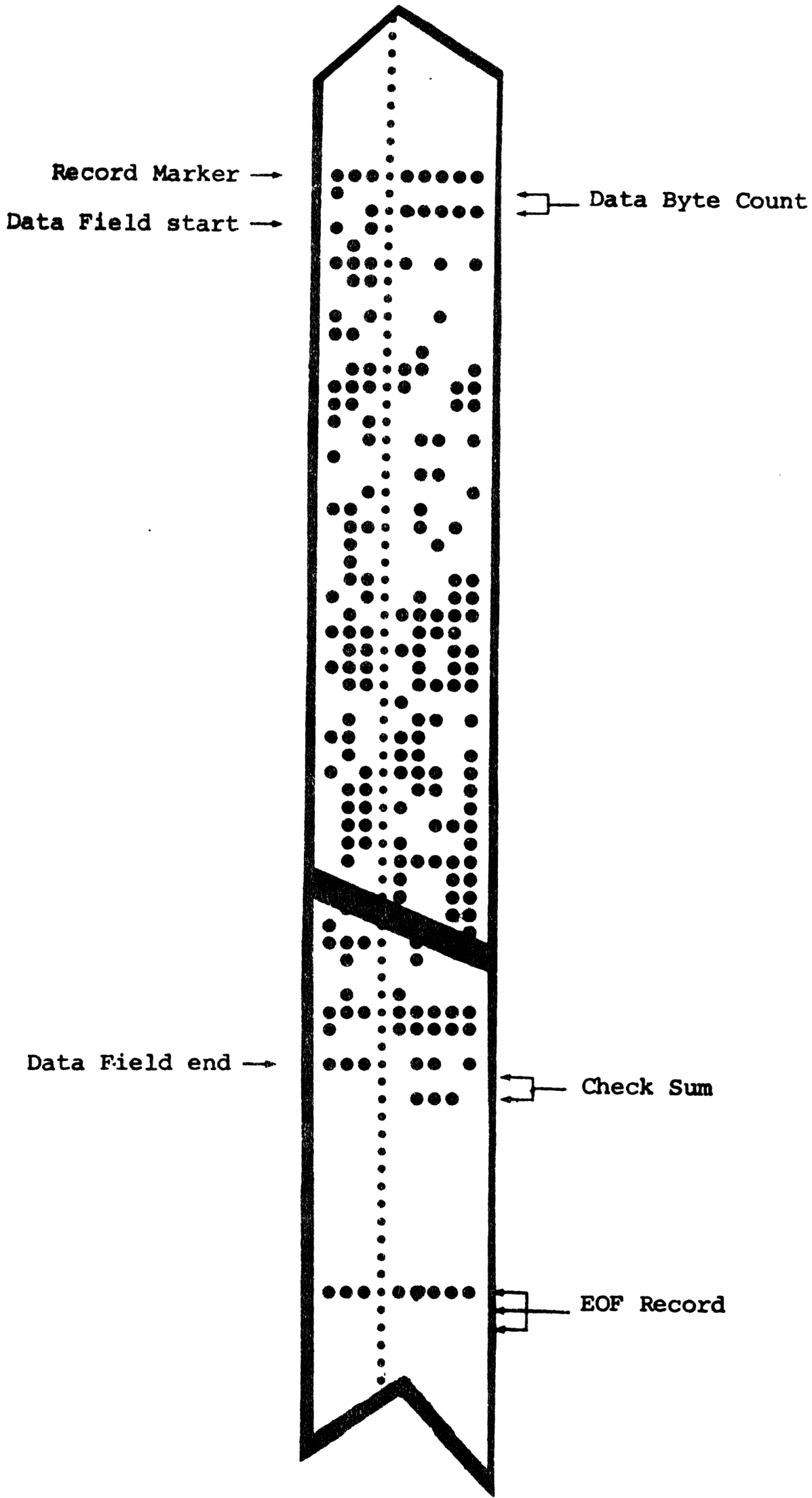
Trailer

The Leader, Gap, and Trailer are each an arbitrary length of unpunched tape. CA software uses these nominal values:

Leader: 8 inches

Gap: 1 inch

Trailer: 9 inches





3. DATA RECORD STRUCTURE

Each Data Record has four fields:

- Record Marker
- Data Byte Count
- Data Field
- Check Sum

Record Marker

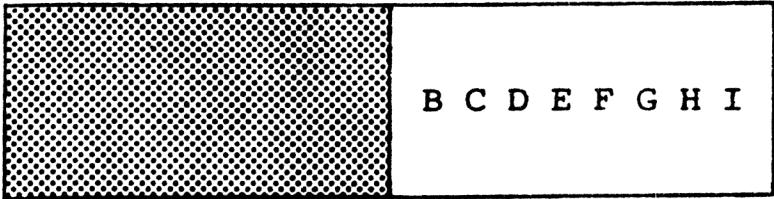
A single frame punched in all 8 channels -- that is, a hexadecimal FF. If the tape is positioned in the Leader or in a Gap, an input routine must read frame-by-frame, and ignore all data until a Record Mark has been reached.

Data Byte Count

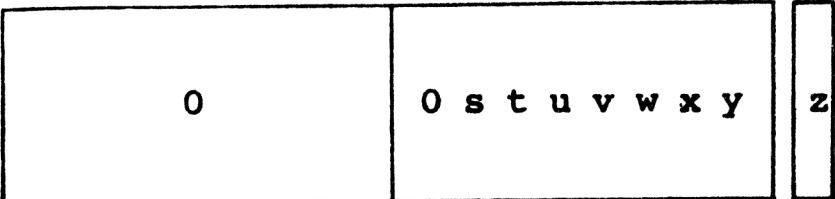
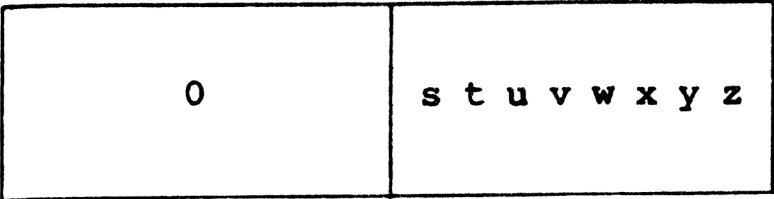
Exactly two frames, containing a 16-bit binary count of the number of frames (or bytes) in the immediately following Data Field.

Data Field

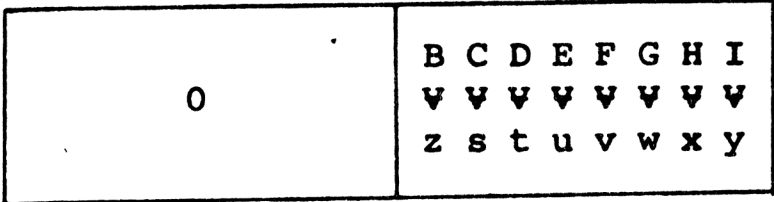
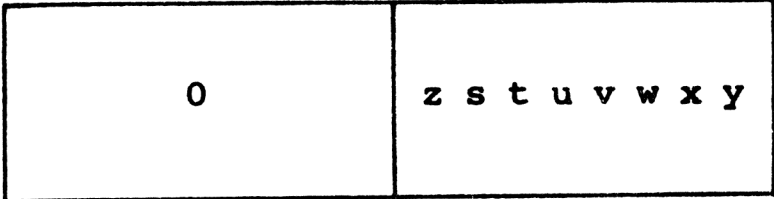
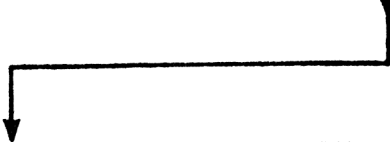
The actual information content of the Data Record appears here. No bit configuration is considered "special" or "control" within the Data Field. When the input routine has exhausted the exact number of frames specified in the Data Byte Count, then the Data Field has ended, and the next two frames will be the Check Sum.



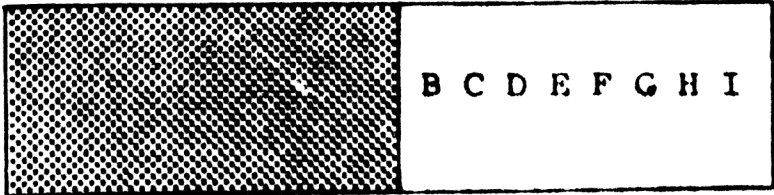
DATA EXCHANGED WITH SUM



SUM ROTATED



EXCLUSIVE-OR SUM WITH DATA



DATA EXCHANGED WITH SUM



4. CHECK SUM

The Check Sum is a cumulative representation of the binary content of both the Data Byte Count and the Data Field itself. It is used to verify the record as a unit, and to detect reading or punching errors somewhere within the record.

The technique used is Right Rotating Exclusive OR Byte Summing. The rotation makes each channel contribute to the sum for every other channel. The Exclusive OR prevents the sum from locking onto a fixed value, regardless of the bit sequences involved. Each byte is successively summed, and the Check Sum itself is a single byte, though it is always punched as two frames, the first of which is blank.

A typical Check Sum algorithm is illustrated on the opposite page. The following code assumes that the byte about to enter the sum is in the low-order 8 bits of Register A. The code does not alter the high-order 8 bits of Register A.

EXCH	A,SUM	EXCHANGE DATA WITH OLD SUM VALUE
SHIFT	A,RO,1	SHIFT SUM RIGHT THRU OV
JF	OV,\$+2	SIMULATE ROTATION OF LOW BYTE ONLY
ADD	=:80,A	
XOR	SUM,A	FORM NEW SUM VALUE IN REG A
AND	=:FF,A	CLEAR NOISE IN HIGH 8 BITS
EXCH	A,SUM	SUM TO MEMORY, DATA BACK TO REG A

5. END-OF-FILE RECORD

An EOF Record is simply a Data Record with a Data Byte Count of exactly zero. No Check Sum is expected for such a record. Visually, an EOF Record is a Record Marker with several inches of blank tape around it.

