~~T77-12925#~~

JSC-12950

80-10286

NASA CR-160667

DESIGN SPECIFICATION
FOR
ERIPS FIELDS DATA BASE DECK CONVERSION

Job Order 81-127

(E80-10286) DESIGN SPECIFICATION FOR ERIPS FIELDS DATA BASE DECK CONVERSION (Lockheed Electronics Co.) 18 p HC A02/MF A01 CSCL 09B

N80-30854

Unclas
G3/43 00286

Prepared By

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Contract NAS 9-12200

FOR

EARTH OBSERVATIONS DIVISION

National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

June 1977

LEC-10775

JSC-12950

DESIGN SPECIFICATION
FOR
ERIPS FIELDS DATA BASE DECK CONVERSION

Job Order 81-127

PREPARED BY

C T Gardner

C. T. Gardner

APPROVED BY

Philip L. Krumm

P. L. Krumm, Supervisor
Applications Software Section

B L Carroll for

T. C. Minter, Supervisor
Techniques Development Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observation Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

June 1977

CONTENTS

1. SCOPE

1.1 GENERAL

This specification establishes the proposed design for a computer program to convert an ERIPS (Earth Resources Interactive Processing System) Fields Data Base (FDB) update card deck to a card deck compatible with input requirements of the Univac 1108 EOD-LARSYS system.

The Requirement Specifications for the program were provided by the Research, Test, and Evaluation (RT&E) Branch of the Earth Observations Division of NASA-JSC.

2. APPLICABLE DOCUMENTS

The following documents, of exact issue shown, form a part of the specification to the extent herein specified.

- Requirements Specification: REF: Interdepartment Communication 643-2042
- IDSD CATEGORY 1 Job Order 81-127, Task Agreement 77-1
- Section 11, Large Area Crop Inventory Experiment (LACIE) User's Guide, Volume 1

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

N/A

3.2 SOFTWARE DESCRIPTION

The purpose of the program is to input the ERIPS (Earth Resources Information Processing System) Fields Data Base (FDB) update card deck and to output (punch) a field definition card deck in the format compatible with the input requirements of the Univac 1108 EOD-LARSYS system of image data processors.

The program will be coded in the IBM 360 Fortran IV language, and will be executable from the LARS/Purdue (Laboratory for Application of Remote Sensing) terminal in JSC Building 17.

3.2.1 SOFTWARE COMPONENT NO. 1 (FDBCVT)

FDBCVT is the main program. The function of FDBCVT is to read the ERIPS Fields Data Base update card deck and to punch an EOD-LARSYS compatible field definition deck for each field defined in the ERIPS card deck. FDBCVT will allow an optional user-input line and/or sample bias to be applied to the input vertex coordinates of each field of a given sample segment in the ERIPS deck, before punching the output EOD-LARSYS field definition deck(s) for the given sample segment.

3.2.1.1 Linkages

FDBCVT calls three subprograms – FIND, NXTCHR, and FIXNUM – to decode the keywords and parameters of the input ERIPS deck.

3.2.1.2 Interfaces

The program is accessed via the LARS/Purdue terminal in JSC Building 17. The interface between the program and the user is the LARS/Purdue IBM 360-67 Control Program (CP) and an associated operating system, the Cambridge Monitor System (CMS). The program-user will utilize the terminal keyboard in building 17 to communicate the appropriate commands to initiate program execution. Operational instructions will be provided in section 4.0 of the Final Design Specifications.

The card reader/punch adjacent to the terminal in Building 17 will be the program's primary input/output interface.

3.2.1.3 Inputs

The input to the Fields Data Base Deck Conversion program, FDBCVT, will consist of an optional BIAS card for each sample segment and an ERIPS Fields Data Base update card deck. The format of the ERIPS deck is given in Section 11, ERIPS User's Guide, Volume 1.

The format of the optional BIAS card is:

CC1	CC11	
BIAS	S=XX	L=YY

The parameters "S=XX" and "L=YY" on the BIAS card contain the user-supplied integers, "XX" and/or "YY", which are additive sample (S) and/or line (L) bias values to be applied to the input ERIPS deck field coordinates.

The BIAS card is optional. If not input, the defaults used by the program are S=0, L=0. Either S or L or both may be input on the BIAS card.

The input ERIPS FDB update deck is the card deck which normally is output (punched) at the LARS/Purdue terminal in Building 17 using the Del-Foster "DEAF" deck as input to a LARS/Purdue program which provides the ERIPS FDB deck as output.

The key words in the ERIPS FDB deck which are expected and responded to by the conversion program, FDBCVT, are:

SEGSTART
FLDSTART
FIELD
CLASS
LINEXX (where XX are numeric)
PIXELXX (where YY are numeric)
FLDEND
SEGEND

Any other key words present in the ERIPS deck are ignored by FDBCVT.

3.2.1.4 Outputs

The FDB deck conversion program, FDBCVT, will provide both line printer and card punch output.

Primary output will be punched cards in a format compatible with the Univac 1108 EOD-LARSYS input requirements. The punched card output will consist of cards in the following formats:

Card type	CC1	CC11
Comment card	COMMENT	SAMPLE SEGMENT ICCCC
Class name card	CLASSNAME	CNAME
Field definition card	FNAME	(1,1), (XXX,YY), (XXX,YYY), (XXX,YYY), (XXX,YYY), *
Field definition continuation card		,(XXX,YYY), (XXX,YYY),···

Printer output provided by the program is as follows:

1. An optional print-out of the input deck

2. An optional print-out of the output (punched) deck

3. Error messages as follows:

a. If an input SEGSTART card cannot be paired with a SEGEND card, the message is

"ERROR--A VALID SEGSTART (SEGEND) CARD BEFORE SEGSTART ID=ICCCC IS MISSING"

b. If the input SEGSTART card is incorrectly formatted (does not have the "=" following "ID") the message is

"ERROR--A SEGSTART CARD IS MISSING AN EQUALS SIGN--LOOK FOR THE NEXT SEGSTART OR EOF"

c. If an input FLDSTART card cannot be paired with a FLDEND card, the message is

"ERROR--A VALID FLDSTART (FLDEND) CARD BEFORE FLDSTART NAME=CCCCCC IS MISSING"

d. If an input FLDSTART card is incorrectly formatted (does not have the "=" following "NAME") the message is

"ERROR--THE FLDSTART CARD CCCCCC IS MISSING AN EQUALS SIGN--LOOK FOR THE NEXT FLDSTART OR SEGEND CARD"

e. If, on the input FIELD cards, a pixel coordinate cannot be paired with its correct line coordinte, the message is

"ERROR--FOR FIELD CCCCCC THE NUMBER OF PIXELS DOES NOT MATCH WITH THE NUMBER OF LINES"

f. If, on the input BIAS card, an "=" is not found following either "S" or "L," the message is

"ERROR IN BIAS CARD--THE EQUALS SIGN IS MISSING FOR EITHER THE SAMPLE AND/OR LINE INCREMENT"

g. When reading the line/pixel coordinates from the FIELD cards if a non-numeric is encountered in a position where a numeric digit is expected (i.e., in the positions occupied by XX or YY in LINEXX = YY or PIXELXX = YY) the message is

"*** CARD IN ERROR IS - FIELD LINEXX = YY
PIXELXX = YY•••"

3.2.1.5 Storage Requirement

TBD

3.2.1.6 Description

The program reads the ERIPS Fields Data Base update deck, card-by-card. The deck may include a user-supplied BIAS card preceding a SEGSTART card. The sample (S) and/or line (L) bias value following the "=" will be added to each input sample and/or line coordinate given on the FIELD card(s) for the given sample segment. The sample/line bias is initialized to zero (0) at the beginning of the program, and at each SEGEND card encountered in the input ERIPS deck. This requires the BIAS card to be present, preceding a SEGSTART card, in order for bias values to be applied to the input field coordinates for a given sample segment. The values input on a BIAS card will be added to each of the sample and line coordinates for all fields defined between the SEGSTART card and the SEGEND card.

For each "SEGSTART ID=ICCCC" card read, the program will punch a LARSYS comment card, "COMMENT SAMPLE SEGMENT ICCCC."

For each "FLDSTART NAME=FNAMEX" card read, the field name following "NAME=" will be the name placed in columns 1-6 of the output field definition cards.

For each set of "FIELD CLASS=CNAMEA LINE01=XX PIXEL01=YY LINE02=XX PIXEL02=YY••• LINEXX PIXELXX" cards read following the "FLDSTART" card, and preceding a "FLDEND" card, the program will output a LARSYS "CLASSNAME CNAMEA" card, followed by EOD-LARSYS field definition cards, with the field name (columns 1-6) from the input FLDSTART card. The output field coordinates will include the bias value(s) from the BIAS card, if input. The format of the output field definition cards is given in Section 3.2.1.4.

The program will continue reading cards from an input ERIPS deck until an end-of-file is encountered.

The punched cards output by the program will be in the Univac FIELDATA character set (i.e., any necessary converson of punched card codes for characters from IBM EBCDIC to Univac FIELDATA is provided by the program).

The format of the input ERIPS deck is expected to be as described in the ERIPS User's Guide, Volume 1, Section 11. The program provides error messages if problems are encountered in interpreting the keywords, separators, or parameters on the input cards. The error conditions and resulting printed messages are described in Section 3.2.1.4.

3.2.1.7 Flowcharts

N/A

3.2.1.8 Program Listing

TBD

3.2.2 SOFTWARE COMPONENT NO. 2 (FIND)

The purpose of the subprogram, Function FIND, is to perform a search for a specific character.

3.2.2.1 Linkages

Function FIND is called by the main program, FDBCVT. Function FIND does not reference any other subprograms.

3.2.2.2 Interfaces

Function FIND interfaces with the calling program via three calling arguments and the function value, which is set within Function FIND.

The function value is set = 1, if a successful character search is completed.

The function value is set = -1 if the character search is unsuccessful.

The calling arguments for FUNCTION FIND are:

ARGUMENT	DIMENSION	TYPE	IN/OUT	DESCRIPTION
CARD	68	A	IN	The input array of 68 words which is assumed to have one character per word, left-justified, blank-filled.
COL	1	I	IN/OUT	On input, the location (word) in CARD, preceding the location at which the search is to begin. On output, the location in CARD at which the character was found. If the character is not found in CARD, COL = initial input value.
VECTOR	1	A	IN	Contains the character to be searched for, left-justified blank-filled in the word.

3.2.2.3 Inputs

The inputs to Function FIND are three calling arguments — CARD, COL, VECTOR — described in Section 3.2.2.2.

3.2.2.4 Outputs

Output from Function FIND is via one calling argument, COL, and the function value which is set within the subprogram (see section 3.2.2.2).

3.2.2.5 Storage Requirements

TBD

3.2.2.6 Description

Function Find performs a search of an input (argument) array, CARD, for the alphanumeric character given in the input argument, VECTOR. The search in CARD will begin at the next location in CARD following the location specified in the input argument, COL. When the specified character is located in CARD, the function value is set = 1, and the location of the character position in CARD is returned in COL. If the search for the specified character is unsuccessful, the function value is set = -1, and COL is returned containing the value it had on entry to Function Find.

3.2.2.7 Flowcharts

N/A

3.2.2.8 Program Listing

TBD

3.2.3 SOFTWARE COMPONENT NO. 3 (NXTCHR)

The purpose of the subprogram, FUNCTION NXTCHR, is to scan a given vector for a non-blank alphanumeric character.

3.2.3.1 Linkages

The subprogram, Function NXTCHR, is referenced by the main program, FDBCVT. The subprograam does not reference any other subprograms.

3.2.3.2 Interfaces

Function NXTCHR interfaces with the calling program via two calling arguments and the function value, which is set within the subprogram.

The function value returned is an alphanumeric character. The character returned is either the first non-blank character found in the input array, CARD, or a "blank" if a non-blank character is not located in CARD.

The calling arguments for Function NXTCHR are:

Argument	Dimension	Type	In/out	Description
CARD	68	A	In	An input array of characters, one character per word, left-justified and blank-filled in each word.
COL	1	I	In/out	On input, COL = the location in CARD preceding the location at which the search for the next non-blank character is to begin. On output, either COL = the location in CARD at which a non-blank character was found, or COL = 67 (the maximum size -1 of CARD) if CARD was all blanks.

3.2.3.3 Inputs

The inputs to Function NXTCHR are two calling arguments — CARD and COL — described in Section 3.2.3.2.

3.2.3.4 Outputs

The output from Function NXTCHR is via the function value and one calling argument, COL (see Section 3.2.3.2).

3.2.3.5 Storage Requirements

TBD

3.2.3.6 Description

Function NXTCHR performs a search of an input (argument) array, CARD, for a non-blank alphanumeric character. The search in CARD will begin at the next location in CARD following the location specified in the input argument, COL. When a non-blank alphanumeric character is found in CARD, the function value is set = the character found, and the location (in CARD) of the character is returned in COL. If a non-blank character is not located in CARD, the function value returned is = "blank," and COL = 67 (the maximum size -1 of CARD).

3.2.3.7 Flowcharts

N/A

3.2.3.8 Program Listing

TBD

3.2.4 SOFTWARE COMPONENT NO. 4 (FIXNUM)

The purpose of the subprogram, Function FIXNUM, is to convert an EBCDIC numeric character to an integer digit.

3.2.4.1 Linkages

Function FIXNUM is called by the main program, FDBCVT. Function FIXNUM does not reference any other subprograms.

3.2.4.2 Interfaces

Function FIXNUM interfaces with the calling program via two calling arguments and the function value, which is set within Function FIXNUM.

The function value returned is the integer resulting from the conversion of the EDCDIC character.

The calling arguments for Function FIXNUM are:

Argument	Dimension	Type	In/out	Description
NUM	1	A	IN	NUM contains the EBCDIC character, left-justified in the word.
MASK	1	A	IN	MASK contains the EBCDIC numeric character "0" (zero), right justified and sign-filled.

3.2.4.3 Inputs

The inputs to Function FIXNUM are two calling arguments — NUM and MASK — described in Section 3.2.4.2.

3.2.4.4 Outputs

The only output of Function FIXNUM is via the function value as set within FIXNUM. The function value returned to the calling program is the integer which results from the conversion of an EBCDIC numeric character.

3.2.4.5 Storage Requirements

TBD

3.2.4.6 Description

Function FIXNUM converts one EBCDIC numeric character input in the calling argument NUM. The conversion of the EBCDIC character to an integer digit is as follows:

1. The input character in NUM is shifted to the right 24 binary positions, resulting in the character being right-justified and the remainder of the word sign-filled (all binary 1's).

2. The right-justified, sign-filled value in MASK (an EBCDIC zero) is subtracted from the right-justified, sign-filled value in NUM.

   The result of the subtraction is an integer, in the range 0-9, if the EBCDIC character in NUM is one of the set, "0", "1", "2", ..., "9".

3. The result of the subtraction is returned as the function value.

   If the result of the subtraction is not an integer in the range 0-9, FIXNUM also outputs a printed message

   "*** ERROR – NUMERIC CHARACTER EXPECTED AND NOT FOUND."

3.2.4.7 Flowcharts

N/A

3.2.4.8 Program Listing

TBD