A Eastern Region Computer Programs
Problems NWS ERCP - No. 15

MDR--PROCESSING MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
November 1983

U.S. DEPARTMENT OF COMMERCE / National Oceanic and Atmospheric Administration / National Weather Service

NOAA Technical Memorandum

National Weather Service, Eastern Region Computer Programs and Problems

The Eastern Region Computer Programs and Problems (ERCP) series is a sub-set of the Eastern Region Technical Memorandum series. It will serve as the vehicle for the transfer of information about fully documented AFOS application programs. The format of ERCP - No. 1 will serve as the model for future issuances in this series.

1 An AFOS version of the Flash Flood Checklist. Cynthia M. Scott, March 1981. (PB81 211252).

2 An AFOS Applications Program to Compute Three-Hourly Stream Stages. Alan P. Blackburn, September 1981. (PB82 156886).

3 PUPPY (AFOS Hydrologic Data Reporting Program). Daniel P. Provost, December 1981. (PB82 199720).

4 Special Search Computer Program. Alan P. Blackburn, April 1982. (PB83 175455).

5 Conversion of ALEMBIC$ Workbins. Alan P. Blackburn, October 1982. (PB83 138313).

6 Real-Time Quality Control of SAOs. John A. Billet, January 1983. (PB83 166082).

7 Automated Hourly Weather Collective from HRR Data Input. Lawrence Cedrone, January 1983. (PB83 167122).

8 Decoders for FRH, FTJ and FD Products. Cynthia M. Scott, February 1983. (PB83 176057).

9 Stability Analysis Program. Hugh M. Stone, March 1983. (PB83 197947).

10 Help for AFOS Message Comp. Alan P. Blackburn, May 1983. (PB83 213561)

11 Stability and Other Parameters from the First Transmission RAOB Data. Charles D. Little, May 1983. (PB83 220475)

12 TERR, PERR, and BIGC: Three Programs to Compute Verification Statistics. Matthew R. Peroutka, August 1983. (PB84 127521)

13 Decoder for Manually Digitized Radar Observations. Matthew R. Peroutka, June 1983. (PB84 127539)

14 Slick and Quick Data Entry for AFOS Era Verification (AEV) Program. Alan P. Blackburn, December 1983.

NOAA Eastern Region Computer Programs and Problems - No. 15

MDR-- PROCESSING MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
November 1983

ACKNOWLEDGEMENTS

Any project of this size must be the work of many, but I must give special thanks to these people.

To Marvin Miller, Meteorologist in Charge, WSFO Cleveland, for his help and encouragement,

To Fred Zuckerberg, Eastern Region Scientific Services Division, who developed the convective Z-R relation, and wrote many of the design specifications,

To the entire staff of the Cleveland Forecast Office, the best guinea pigs in the region, for their help and patience,

To Mary Newton, Techniques Development Laboratory, who sent me her first rendition of this program,

To Alan Rezek and Harvey Thurm, Eastern Region DATAC, who would not leave me alone until I began this project, and

To Melvina McDonald who supplied the cartographic routines for handling the national MDR grid.

MDR--PROCESSING MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
WSFO Cleveland, Ohio

I. General Information

A. Summary

National Weather Service hourly radar observations contain manually digitized information which is often not used by field people. Plotting and analyzing data of this sort from a number of sites can be very tedious and time-consuming. The reports are merged into the national radar maps which are transmitted through facsimile and AFOS. These national charts, however, lack resolution, and they are generally not available until an hour after the observations are taken.

This study describes a program which can cull this digitized data from a number of radar observations, and then put them into a usable form. Three graphic products are generated representing one hour, three hours, and six hours of digitized data.

Four different formats are available. The first uses a convective Z-R relation to generate rainfall amounts. This relation is described in Appendix I (It accounts for the average areal coverage of the various VIP levels. Initial testing in the northeastern United States suggests that rainfall totals are overestimated by a factor of two.). The second relation is used for stratiform rain, and it was taken from the National Weather Service Radar Users Guide. Local users can develop their own relation, and this is the third option. The fourth relation simply totals the MDR values.

B. Environment

The programs are written in Data General's FORTRAN IV, and run in the background partition of the Eclipse mini-computer. The local database must contain the necessary radar observations, and it must contain the graphic products for output.

C. References

Brandes, Edward A. "The Variations of Oklahoma Spring Rains as Revealed by Radar." Preprints of the Eighth Conference on Severe Local Storms. October, 1973.

National Weather Service. Radar Code User's Guide. National Weather Service, Silver Spring, Maryland. December, 1980.

MacDonald, Alexander E. AFOS Graphic Creation from FORTRAN. NOAA Western Region Computer Programs and Problems, NWS WRCP-18. Salt Lake City, Utah. March, 1981.

Peroutka, Matthew R. A Decoder for Manually Digitized Radar Observations. NOAA Eastern Region Computer Programs and Problems, NWS ERCP-13. June, 1983.

II. Application

A. Complete Program Description.

Figure 1 is a flow diagram for the program MDR.SV.

The subroutines MDRUN and MDRSU set up housekeeping for the program. MDRUN reads the file COM.CM and interprets the global and local switches. MDRSU reads the MDRIT file (Information Table). The convective and stratiform Z-R relations are also stored in this subroutine in DATA statements.

The program then enters a loop which decodes the observations (using ROBDEC) and then transfers the data into the MDR array (using MDRGRID). The decoding routines have already been described iN ERCP-13 (Peroutka, 1983). If any problems are encountered in decoding an observation from a network radar, an alert is sent to the console which initiated the program. The MDRGRID routine must decide which VIP level to accept when two observations disagree at a common grid square. The priority used is as follows: 6, 5, 4, 3, 2, 1, 8 (intensity unknown, but severe weather suspected), 9 (intensity unknown), and 7 (missing data).

Once this loop is complete, the MDR array should be filled with seven hours of MDR data. Before this data is converted to its final form, subroutine MDRCND is called to "condition" the array. If a VIP level is missing for a given point at hour H, but the data are available for H-1 and H+1, an average value is inserted. If hour one is missing, but hour two is available, persistence is assumed. Likewise, if hour seven is missing, but hour six is available, then the VIP level from hour six is inserted.

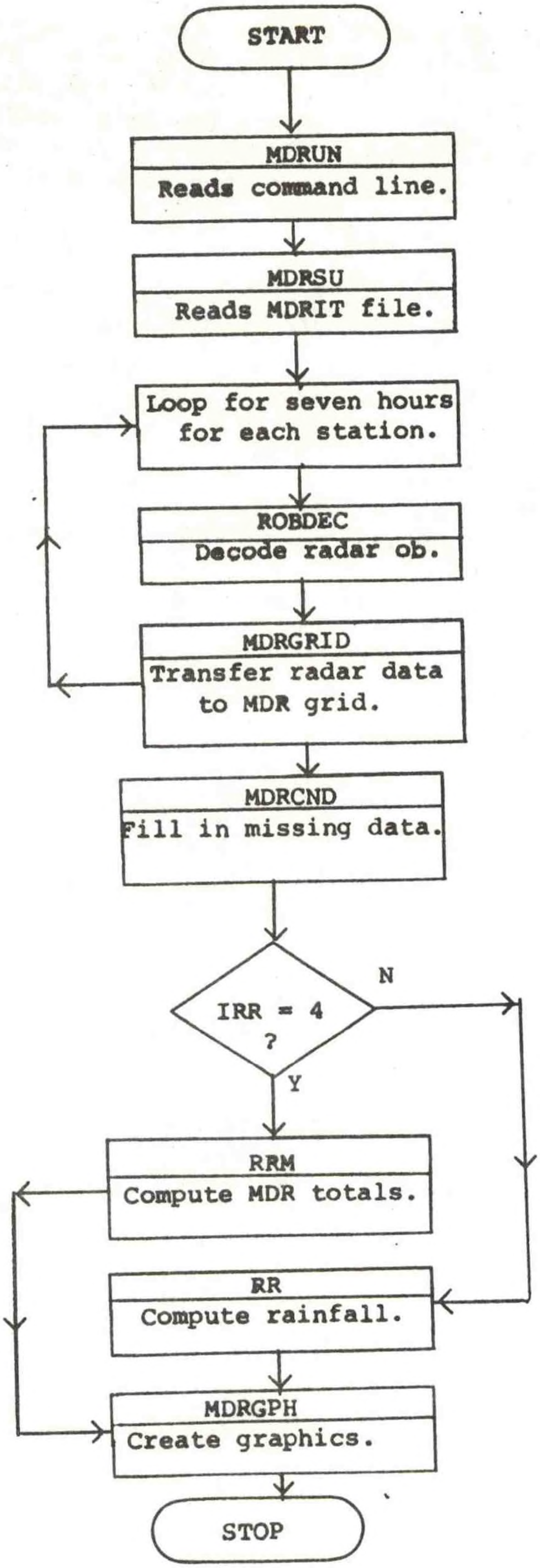

Figure 1. Flow Chart for MDR.SV

There are two routines which convert the VIP levels in the MDR array into rainfall data in the IRFL array. The subroutine RR uses the three Z-R relations to compute rainfall in hundreths of an inch. To smooth the data over time, the rainfall rate for a given hour is computed from the arithmetic mean of the two observations on either side of the hour. This means that -

RR1 = (R1+R2)/2,

RR3 = (R1+R2)/2 + (R2+R3)/2 + (R3+R4)/2, and

RR6 = (R1+R2)/2 + (R2+R3)/2 + (R3+R4)/2 + (R4+R5)/2 + (R5+R6)/2 + (R6+R7)/2, where

RR1 = one-hour rainfall rate,
RR3 = three-hour rainfall accumulation,
RR6 = six-hour rainfall accumulation, and
Rn = rainfall rate associated with observation n.

For computational efficiency, these equations reduce to the following:

RR1 = (R1+R2)/2,

RR3 = (R1+R4)/2 + R2 + R3, and

RR6 = RR3 + (R4+R7)/2 + R5 + R6.

The three highest-order bits of each word in the array IRFL are reserved for incomplete data (1B0), level five or six (1B1), and future expansion. Levels eight and nine qualify as missing data.

The subroutine RRM simply totals the VIP levels into the IRFL array, again keeping track of missing data and levels five and six.

The final task of the program is to convert the IRFL array into a usable format. The subroutine MDRGPH creates three graphic products, NMCGPHRR1, NMCGPHRR3, and NMCGPHRR6. The AFOS Graphic Library routines (MacDonald, 1981) are used to produce the charts. The subroutine MDRLB generates the labels for the products.

Temporary files (RR1, RR3, and RR6) are then deleted from the disk, and an alert is sent to the console with originated the program.

B. Machine Requirements

The program MDR.SV runs in about 15K words of core, and it takes up about 54 blocks of disk space. The file MDRIT must be available, but it only requires one block. The program MDRGEN.SV runs in less than 10K words of core. The file is 22 blocks in length, but it is only needed once when the program is set up. Most sites will be able to leave MDRGEN.SV on floppy diskette. Three temporary files are created on the disk by the graphic routines. They are deleted at the end of the program, and they should only require a couple of blocks each.

Execution time depends on the number of observations which are decoded, and the size of the area of interest. One or two minutes is a fair estimate; if any of the observations are on the wish list, however, the program seems to slow a great deal.

C. Database

The desired radar observations should be stored in the local database. Although only seven hours of data are used in any one run, more versions should be stored to account for special observations. If the database holds enough versions, it is possible to analyze a weekend storm on Monday morning. The observations should be version purged, not time purged.

The three graphics NMCGPHRR1, NMCGPHRR3, and NMCGPHRR6 must be in the database with an appropriate map background set. (The current program can only handle backgrounds one, two, and three. Future enhancements call for many other backgrounds to be developed. At present, backgrounds two and three work best.)

III. Procedures

A. Installation

1. Figure 2 is the overlay map which is used with the teletype plots of MDR data (the "paper doll" messages). This map can also be found in the Radar Code User's Guide. (It is on page twenty-one in the current manual.)

2. Decide on a rectangular area of interest. The size of this rectangle has a great effect on program run time. The current program limits this area to a thirty-by-thirty grid. Find the coordinates (MDR grid boxes) of the upper left and lower right corners.

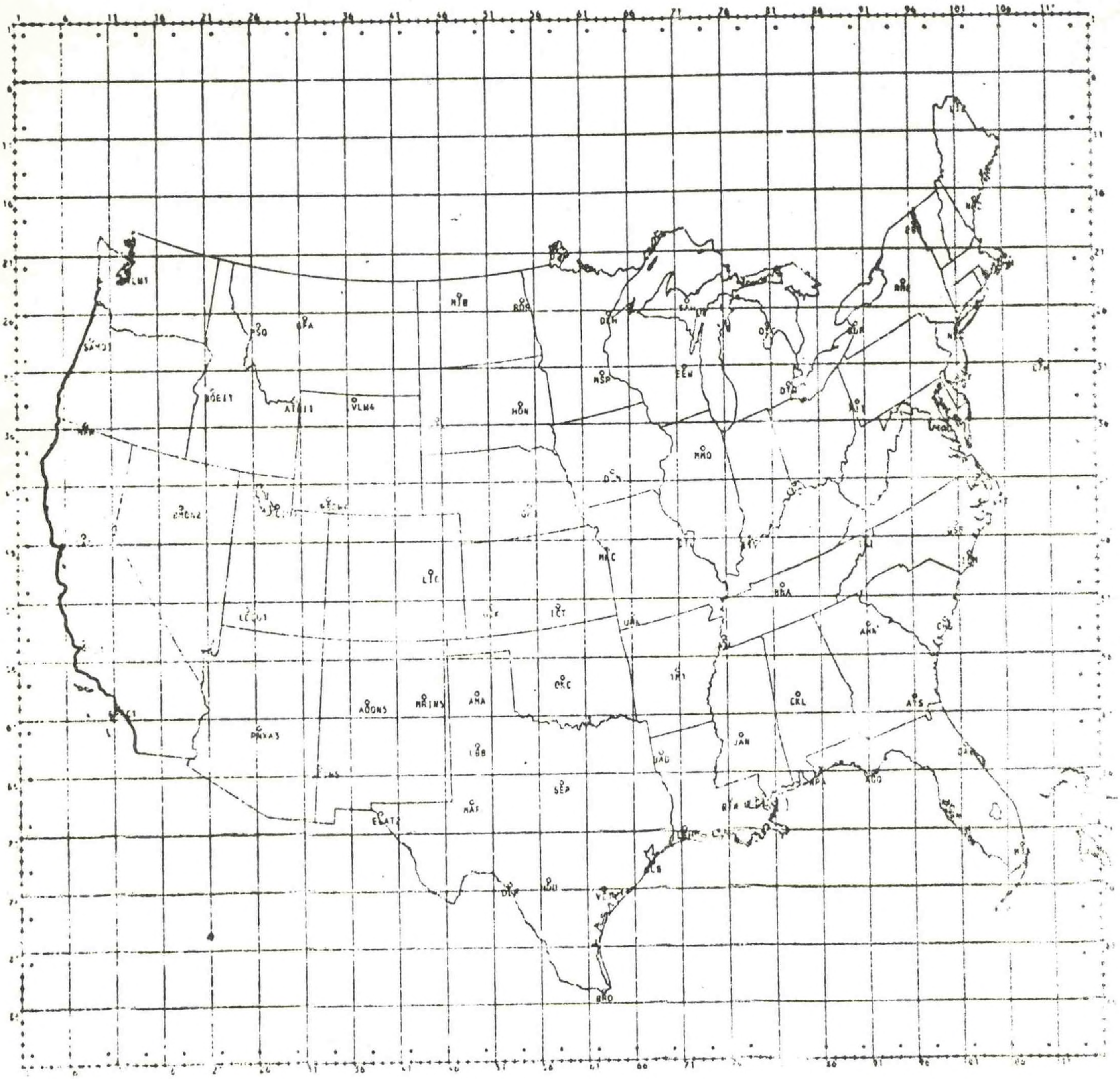

Figure 2.: Map of the MDR grid. I-coordinates are on the vertical. J-coordinates are on the horizontal.

3. Decide which radar sites will be used. The best way to do this is to flip through the maps in the back of the Radar Code User's Guide. Take note of the radars you want to use and whether they are network radars. (Don't take a radar unless it lies within seven grid squares of the area of interest.)

4. Get the coordinates of each radar from the Tables 1 and 2.

5. Make sure all of the products are in the database.

6. If you want to define an optional Z-R relation, work it up now. Don't forget about VIP levels eight and nine. Enter the rainfall rate for each VIP level without a decimal point.

7. Run the program MDRGEN.SV to generate the data file MDRIT. The program proceeds with a rather simple dialogue. Figure 4, Appendix II contains a sample run. If you want to change the parameters in MDRIT, you can re-run MDRGEN; it will delete the old MDRIT file. If you are handy with the octal editor, figure three contains the format for MDRIT.

8. If you want to alter the convective or stratiform relations, they are stored in the common block Z2R. Convective runs from 2270 to 2301 (VIP levels zero through nine). Stratiform runs from 2302 to 2313.

B. Initiation

Once the database is ready and MDR.SV and MDRIT exist on DPØ (at least as links), the program is ready to run. The simplest command is:

RUN:MDR

at an ADM. This will produce one-, three-, and six-hour rainfall data ending with the current hour. The default rainfall relation is set by MDRGEN.

Four global switches allow the user to choose between the various Z-R relations. They are-

/C for the convective relation,
/M for MDR totals,
/O for the locally-derived optional relation, and
/S for the stratiform relation.

If a different span of time is needed, three local switches can also be used. They are-

TABLE 1

NETWORK RADARS AND THEIR COORDINATES

	I	J
Alliance, NE	40	46
Amarillo, TX	59	49
Apalachicola, FL	66	91
Athens, GA	53	91
Atlantic City, NJ	31	101
Binghamton, NY	26	95
Bristol, TN	46	91
Brownsville, TX	84	62
Brunswick, ME	16	103
Buffalo, NY	27	90
Cape Hatteras, NC	42	105
Centreville, AL	59	83
Charleston, SC	53	99
Chatham, MA	21	106
Cincinnati, OH	41	83
Daytona Beach, FL	63	101
Des Moines, IA	40	64
Detroit, MI	32	82
Evansville, IN	46	78
Fargo, ND	27	56
Galveston, TX	73	67
Garden City, KS	51	51
Grand Island, NE	43	55
Hondo, TX	75	56
Huron, SD	34	54
Jackson, MS	62	77
Kansas City, MO	46	63
Key West, FL	77	105
Lake Charles, LA	70	71
Limon, CO	48	44
Little Rock, AR	57	70
Longview, TX	65	66
Marseilles, IL	38	73
Medford, OR	35	8
Memphis, TN	54	75
Miami, FL	72	107
Midland, TX	68	48
Minneapolis, MN	31	63
Missoula, MT	27	26
Monett, MO	52	66
Nashville, TN	50	82
Neenah, WI	31	71
Oklahoma City, OK	57	58

Table 1 (Cont.)

	I	J
Patuxent River, MD	36	99
Pensacola, FL	66	85
Pittsburgh, PA	34	90
Sacramento, CA	45	7
Slidell, LA	68	79
Stephenville, TX	66	58
Saint Louis, MO	45	71
Tampa, FL	69	99
Volens, VA	41	96
Waycross, GA	59	96
Wichita, KS	51	58
Wilmington, NC	47	102

TABLE 2

LOCAL WARNING RADARS AND THEIR COORDINATES

	I	J
Abilene, TX	66	54
Akron, OH	34	87
Albany, NY	23	98
Alpena, MI	26	79
Atlanta, GA	55	88
Augusta, GA	53	94
Austin, TX	72	60
Baton Rouge, LA	68	78
Beckley, WV	41	91
Billings, MT	31	35
Bismarck, ND	28	49
Burlington, VT	18	97
Charleston, WV	40	89
Charlotte, NC	48	94
Cheyenne, WY	43	42
Cleveland, OH	33	86
Columbia, MO	46	68
Columbia, SC	51	95
Columbus, GA	58	88
Columbus, OH	37	85
Concordia, KS	46	57
Corpus Christi, TX	79	61
Duluth, MN	26	63
Erie, PA	30	88
Fort Smith, AR	56	65
Fort Wayne, IN	37	80
Goodland, KS	47	48
Harrisburg, PA	32	96
Hartford, CT	23	101
Houghton Lake, MI	28	78
Huntsville, AL	54	82
Indianapolis, IN	40	79
Jackson, KY	44	87
Las Vegas, NV	54	19
Los Angeles, CA	58	11
Louisville, KY	44	81
Lubbock, TX	63	49
Macon, GA	57	91
Madison, WI	34	71
Marquette, MI	24	71
Meridian, MS	61	80
Mobile, AL	66	82
Moline, IL	38	70
Montgomery, AL	60	85
Muskegon, MI	32	76

Table 2 (Cont.)

	I	J
Norfolk, Ne	40	56
North Platte, NE	43	50
Omaha, NE	41	59
Phoenix, AZ	63	25
Portland, OR	27	11
Raleigh, NC	44	98
Rapid City, SD	35	45
Rochester, MN	33	64
San Angelo, TX	69	53
San Juan, PR	69	155
Shreveport, LA	64	68
Sioux Falls, SD	35	57
Springfield, IL	42	72
Topeka, KS	47	61
Tulsa, OK	55	62
Victoria, TX	76	62
Waco, TX	68	60
Waterloo, IA	37	65
West Palm Beach, FL	69	106
Wichita Falls, TX	61	57
Williston, ND	25	44
Worcester, MA	22	102

MDRIT FILE FORMAT

Missing data are replaced be zeros.

WORD	DESCRIPTION
000 - 004	CCCNNNXXX for product one (null in last byte)
005	Special function switches: 1B15 = network radar
006	Offset to convert station I-coordinate to regional I-coordinate
007	Offset to convert station J-coordinate to regional J-coordinate
010 - 017	Same as 000 to 007 for product two
020 - 027	Same as 000 to 007 for product three
.	.
.	.
.	.
230 - 237	Same as 000 to 007 for product twenty
240	I-coord of upper left corner of regional grid
241	J-coord of upper left corner of regional grid
242	Height of regional area in grid squares
243	Width of regional area in grid squares
244	VIP level 1 rainfall (in hundreths)
245	VIP level 2 rainfall
246	VIP level 3 rainfall
247	VIP level 4 rainfall
250	VIP leavel 5 rainfall
251	VIP level 6 rainfall
252	VIP level 8 rainfall
253	VIP level 9 rainfall
254	Map background number
255	Default rainfall relation (1=convective, 2=stratiform, 3=optional, 4=MDR totals)
256 - 377	Future expansion

Figure 3: MDRIT file format

/M	to set the month,
/D	to set the day, and
/H	to set the hour.

Some examples.

RUN:MDR 18/H	default relation, 12Z to 18Z today.
RUN:MDR/C 17/D 12/H	convective relation, 6Z to 12Z on the 17th.
RUN:MDR/M 10/M 5/D 6/H	MDR totals, 0Z to 6Z on October 5th.

C. Output

The graphic products use the following station model at each MDR grid square:

```
R R
M *
```

For MDR totals, "RR" represents the MDR total in tens and units. For the other relations, "RR" represents inches and tenths of rainfall. (On most map backgrounds, there is no room for the decimal point.) The "M" appears only if data were missing when the rainfall (or MDR total) was computed. The asterisk appears if a level five or six was encountered during the computation. An example of one graphic product (RR6) appears in Figure 5.

A label at the bottom of the field shows the ending date and time and which relation was used.

The program can generate two alerts. The first warns that there was a problem decoding an observation from a network radar. This usually occurs when the program is run before the latest observations are received, but it can also indicate coding problems within the observations. The program will continue. Usually surrounding radar sites will provide the needed information but in certain cases where there is no overlap, data may be missing.

The second alert indicates the program has finished.

D. Cautions

As with any decoder, if the input ROB is sufficiently garbled the MDR decoder may fail.

The MDR decoder checks the date-time polynomial, the date in the WOUS00 line, and the time in the body of the observation. If an observations is miscoded or missing, it is almost impossible to "bogus" it into the database.

The convective Z-R relation makes some very important assumptions about the size of the various VIP levels, and the size of an MDR grid square. If these assumptions are not valid (pinpoint level two's or large level four's for example), the rainfall estimates will be in error.

The program makes no attempt to process the contractions MALF or PALF. This can produce large errors in dry situations. Similarly, when the freezing level is low, wet snow aloft can increase reflectivity drastically.

There may be cases of missing observations which are not coded as "M" on the graphic, e.g. distant radar provides an under-estimate to a grid square common to a closer radar whose observation is missing. ADM error messages should be noted and the affected portion of the MDR grid should be evaluated more closely.

IV. Program Listings

```
C               MDR.SV                      MONITORS RADAR OBSERVATIONS.
      DIMENSION IDA(3), IDTMS(4,7), IGEN(0:255), ISTAR(13,13)
      DIMENSION KEY(20), MDR(30,30,7), IRFL(30,30,3)
C
C      THIS PROGRAM READS RADAR OBSERVATIONS FROM THE DATABASE AND GENERATES
C      THREE GRAPHICS WHICH SHOW CURRENT RAINFALL RATE, THREE-HOUR RAINFALL
C      ACCUMULATION, AND SIX-HOUR RAINFALL ACCUMULATION.
C
      CALL MDRUN(IDA,IHR,IRR)              ; GET DATA FROM COMMAND LINE.
      CALL MDRSU(IDA,IHR,IRR,IDTMS,IGEN)   ; SET UP ARRAYS.
      IMAX = IGEN(162)
      JMAX = IGEN(163)
      DO 100 I = 1, IMAX
      DO 100 J = 1, JMAX
      DO 100 K = 1, 7
  100 MDR(I,J,K) = 7
      IPTR = -8                            ; LOOP FOR EACH STATION.
  200 IPTR = IPTR + 8
      IF (IGEN(IPTR).EQ.0.OR.IPTR.EQ.160) GOTO 600
      CALL KSRCF(IGEN(IPTR),KEY,IER)       ; GET KEY RECORD FOR THIS STATION.
      NVER = IAND(KEY(9),177K)             ; GET NUMBER OF VERSIONS STORED.
      IOFF = IGEN(IPTR+6)                  ; OFFSETS TO GET TO REGIONAL GRID.
      JOFF = IGEN(IPTR+7)
      IR = 0
      DO 500 I = 1, 7                      ; LOOP FOR THE SEVEN HOURS NEEDED.
       DO 300 J = 1, 3                     ; SET UP DATE AND TIME FOR OB NEEDED.
  300  IDA(J) = IDTMS(J,I)
       IHR = IDTMS(4,I)
       CALL ROBDEC(IDA,IHR,NVER,ITYP,ISTAR)    ; DECODE RADAR OBSERVATION.
       IF (ITYP.NE.-1) GOTO 400            ; DECODER FAILURE AT A NETWORK SITE.
        IF (IAND(IGEN(IPTR+5),1).NE.1) GOTO 350
        IDA(1) = KEY(4)
        IDA(2) = IOR(IAND(177400K,KEY(5)),40K)
        IDA(3) = 0
        IF (IR.EQ.0) CALL FORKE("DECODE ROB",IDA,IER)
        IR = 1
  350   CALL KSRCF(IGEN(IPTR),KEY,IER)     ; GO BACK TO FIRST VERSION.
        GOTO 500
  400  IF (ITYP.EQ.2.OR.ITYP.EQ.3) GOTO 500    ; DON'T GRID A MISSING OB.
       CALL MDRGRID(MDR,IMAX,JMAX,I,ISTAR,IOFF,JOFF)     ; GRID THE OB.
  500 CONTINUE
      GOTO 200                             ; GET THE NEXT STATION.
  600 CALL MDRCND(MDR,IMAX,JMAX)           ; "CONDITION" THE MDR GRID.
      IF (IRR.NE.4) CALL RR(MDR,IRR,IRFL,IMAX,JMAX)   ; CNV MDR DATA TO RAFL.
      IF (IRR.EQ.4) CALL RRM(MDR,IRFL,IMAX,JMAX)      ; MDR TOTALS.
      CALL MDRGPH(IDTMS,IGEN,IRR,IRFL,IMAX,JMAX)  ; GENERATE THE GRAPHICS.
      CALL WAIT(5,2,IER)
      CALL DFILW("RR1",IER)
      CALL DFILW("RR3",IER)
      CALL DFILW("RR6",IER)
      CALL FORKP("MDR","RR1 RR3 & RR6",IER)
      CALL EXIT
      END
```

```
      SUBROUTINE MDRUN(IDATE,IHR,IRR)
      DIMENSION IDATE(3), IBF(10)
      LOGICAL BTEST
C
C      THIS SUBROUTINE READS THE COMMAND LINE AND RETURNS THE DATE (IDATE)
C      AND HOUR TO PROCESS ALONG WITH THE RAINFALL RELATION (IRR) TO USE.
C      THE DEFAULTS ARE RELATION 0 AND THE CURRENT DATE AND TIME.
C
C            RUN:MDR/C 5/M 24/D 19/H     ===> CONV., MAY 24 AT 1935Z
C            RUN:MDR/S 10/M 05/D 12/H    ===> STRAT., OCT 5 AT 1235Z
C            RUN:MDR/O 18/H              ===> OPTIONAL, TODAY AT 18Z
C
      CALL DATE(IDATE,IER)                            ;SET UP DEFAULTS.
      CALL FGTIM(IHR,I,J)
      IF (I.LT.35) IHR = IHR - 1
      IF (IHR.GE.0) GOTO 100
       CALL YDA(IDATE)
       IHR = 0
  100 IRR = 0
      CALL GCHN(IC,IER)                               ;READ COM.CM.
      CALL OPENR(IC,"COM.CM",0,IER)
      CALL RDL(IC,IBF,N,IER)                          ;CHECK GLOBAL SWITCH.
      IF (IER.NE.1) GOTO 300
      N = 4
      CALL RDS(IC,IBF,N,IER)
      IF (BTEST(IBF(1),13)) IRR = 1                   ;CONVECTIVE.
      IF (BTEST(IBF(2),13)) IRR = 2                   ;STATOFORM.
      IF (BTEST(IBF(1),1)) IRR = 3                    ;OPTIONAL.
      IF (BTEST(IBF(1),3)) IRR = 4                    ;MDR TOTALS.
  200 CALL RDL(IC,IBF,N,IER)                          ;PROCESS LOCAL SWITCHES.
      IF (IER.NE.1) GOTO 300
      I = ISHFT(IBF(1),-8) - 48
      IF (N.EQ.3.OR.N.EQ.5) I = I*10 + IAND(IBF(1),377K) - 48
      N = 4
      CALL RDS(IC,IBF,4,IER)
      J = IBF(1)
      IF (BTEST(J,12)) IDATE(2) = I                   ;DAY.
      IF (BTEST(J,8)) IHR = I                         ;HOUR.
      IF (BTEST(J,3)) IDATE(1) = I                    ;MONTH.
      GOTO 200
  300 CALL KLOSE(IC,IER)
      RETURN
      END
```

```
      SUBROUTINE MDRSU(IDATE,IHR,IRR,IDTMS,IBF)
      DIMENSION IDATE(3), IDTMS(4,7), IBF(0:255)
      COMMON /Z2R/ IV2R(10,3)
      DATA IV2R/ 0,10,47,63,77,84,85,0,0,0,
     +0,5,30,75,77,84,85,0,0,0,10*0/
C
C      THIS SUBROUTINE GENERATES THE SEVEN DATE-TIME GROUPS, READS THE
C      MDRIT FILE, AND (IF IT WILL BE NEEDED) LOADS THE OPTIONAL RAINFALL
C      RELATION INTO IV2R.
C
      DO 200 I = 1, 7                                 ;SET UP DATE-TIME ARRAY.
       DO 100 J = 1, 3
  100  IDTMS(J,I) = IDATE(J)
       IDTMS(4,I) = IHR
       IF (IHR.GT.0) GOTO 200
        CALL YDA(IDATE)
        IHR = 24
  200 IHR = IHR - 1
      CALL GCHN(IC,IER)                               ;READ MDRIT.
      CALL OPENR(IC,"MDRIT",0,IER)
      CALL RDB(IC,IBF,0,1,IER)
      CALL ERROR(IER,"CAN'T READ MDRIT")
      CALL KLOSE(IC,IER)
      IF (IRR.EQ.0) IRR = IBF(173)                    ;GET DEFAULT RELATION.
      IF (IRR.NE.3) GOTO 400
      DO 300 I = 1, 10                                ;LOAD OPTIONAL RAINFALL
       J = 163 + I                                    ;RELATION.
       IV2R(I+1,3) = IBF(J)
  300 CONTINUE
      IV2R(9,3) = IBF(170)
      IV2R(10,3) = IBF(171)
  400 RETURN
      END
```

```
      SUBROUTINE ROBDEC(IDA,IHR,NVER,ITYP,MDRAR)
      DIMENSION IDA(3), MDRAR(13,13), IBF(128), IUP(512)
C
C      THIS SUBROUTINE SEARCHES VERSIONS OF THE CURRENT PRODUCT TO FIND A ROB
C      FOR THE DATE (IDA) AND TIME (IHR) REQUESTED.  THE MDR SECTION OF THE OB
C      IS USED TO FILL THE 13X13 ARRAY (MDRAR) WITH MDR VAULUES.  ITYP =
C
C      -1 FOR FAILURE    0 FOR NO PROBLEMS    1 FOR PPINE    2 FOR PPIOM
C       3 FOR PPINA      4 FOR ROBEPS         5 FOR ARNO     6 FOR RHINO.
C
      CALL GROB(IDA,IHR,NVER,IBF,IUP,I)             ;CHECK DATES AND TIMES.
      IF (I.EQ.-1) GOTO 500
      IEND = MASK("<0><203>",1,IUP,I,256)          ;SEARCH FOR END OF PRODUCT
      IF (IEND.NE.-1) GOTO 100                     ;IN FIRST BLOCK.
       CALL RDBKF(1,IBF,IER)                       ;LOAD SECOND BLOCK.
       IF (IER.NE.1) GOTO 500
       CALL UNPACK(IBF,256,IUP(257))
       IEND = MASK("<0><203>",1,IUP,257,512)
       IF (IEND.EQ.-1) IEND = 512
  100 CALL ROBOP(IUP,I,IEND,ITYP)                  ;SEARCH FOR OPERATIONAL
      IF (ITYP.EQ.2.OR.ITYP.EQ.3) GOTO 550         ;CONTRACTIONS.
      DO 200 I1 = 1, 13                            ;ZERO OUT THE ARRAY.
       DO 200 J1 = 1, 13
  200 MDRAR(I1,J1) = 0
      IF (ITYP.EQ.1) GOTO 550
      I = MASK("<0><136>",1,IUP,I,IEND)            ;SEARCH FOR ^.
      IF (I.EQ.-1) GOTO 500
      IE = MASK("<0><75>",1,IUP,I,IEND)            ;SEARCH FOR =.
      IF (IE.NE.-1) IEND = IE
      I = I + 1
      IEND = IEND - 1
  300 I = MASK("AZAZ09",3,IUP,I,IEND)              ;SEACH FOR THE PATTERN
      IF (I.EQ.-1) GOTO 550                        ;"LETTER-LETTER-NUMBER".
      I1 = IUP(I) - 70                             ;ANALYZE THE TWO LETTERS.
      J1 = IUP(I+1) - 70
      DO 400 J = 2, 26                             ;NOW LOOK AT THE CHARACTERS
       K = IUP(I+J) - 48                           ;FOLLOWING.
       IF (K.GE.0.AND.K.LE.9) GOTO 350             ;  IS IT A NUMBER?
        I = I + J                                  ;  NO.  SEARCH AGAIN.
        GOTO 300
  350  J2 = J1+J-2                                 ;  YES.  LOAD IT INTO THE
       IF (J2.GE.1.AND.J2.LE.13) MDRAR(I1,J2) = K  ;  MDR ARRAY.
  400 CONTINUE
      I = I + 14
      GOTO 300
  500 ITYP = -1                                    ;FAILURE.
  550 RETURN
      END
```

```
      SUBROUTINE GROB(IDA,IHR,NVER,IBF,IUP,J)
      DIMENSION IDA(3), IBF(128), IUP(512)
C
C      THIS SUBROUTINE CHECKS ALL VERSIONS OF THE CURRENT PRODUCT TO FIND A
C      ROB WITH THE REQUESTED DATE AND TIME.  ON RETURN, J POINTS TO THE FIRST
C      DIGIT OF THE TIME IN IUP.  J = -1 INDICATES AN ERROR.
C
      JD = JDATE(IDA(1),IDA(2),IDA(3))
      CTIML = (JD-1)*1440. + (IHR-2)*60.                  ;THE WINDOW IN JULIAN
      CTIMU = CTIML + 240.                                ;MINUTES.
      DO 500 I = 1, NVER                                  ;LOOP TO CALL UP PREVIOUS
       CALL RDBKF(0,IBF,IER)                              ;VERSIONS.
       IF (IER.NE.1) GOTO 700
       CALL UNPACK(IBF,256,IUP)
       PTIM = IUP(17)*16384. + IUP(18)*128. + IUP(19)     ;CREATION TIME.
       IF (PTIM.LT.CTIML) GOTO 700                        ;TOO OLD.  ERROR.
       IF (PTIM.GE.CTIMU) GOTO 400                        ;TOO NEW.  TRY PRVS VERSN.
       J = MASK("<0> <0>K0Z0Z0Z<0> ",6,IUP,30,40)         ;SEARCH FOR " KCCC ".
       IF (J.EQ.-1) GOTO 400
       J = J + 6                                          ;J NOW POINTS TO DAY.
       IPD = (IUP(J) - 48)*10 + IUP(J+1) - 48
       IF (IPD.EQ.IDA(2)) GOTO 100                        ;RIGHT DAY.
       JD1 = JD + 1                                       ;CHECK FOR 00Z CROSSING.
       CALL J2MDA(JD1,M,ND,IDA(3))
       IF (ND.NE.IPD) GOTO 400
100    J = J + 6                                          ;SEARCH FOR TIME IN ROB.
       J = MASK("<0> 09090909",5,IUP,J,J+20)
       IF (J.EQ.-1) GOTO 400
       J = J + 1                                          ;CHECK THIS TIME.
       IPH = (IUP(J) - 48)*10 + IUP(J+1) - 48
       IF (IPH.EQ.IHR) GOTO 750
400    CALL PRVRF(IER)                                    ;GET PREVIOUS VERSION.
       IF (IER.NE.1) GOTO 700
500   CONTINUE
700   J = -1
750   RETURN
      END
```

```
      SUBROUTINE ROBOP(IUP,IBGN,IEND,ITYP)
      DIMENSION IUP(512)
C
C      THIS SUBROUTINE SEARCHES THE RADAR OBSERVATION UNPACKED IN IUP FROM
C      IBGN TO IEND FOR OPERATIONAL CONTRACTIONS.  ITYP IS SET TO:
C
C      0 FOR NO CONTRACTIONS    1 FOR PPINE    2 FOR PPIOM    3 FOR PPINA
C      4 FOR ROBEPS             5 FOR ARNO     6 FOR RHINO.
C
      ITYP = 1
      IF (MASK("<0>P<0>P<0>I<0>N<0>E",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 2
      IF (MASK("<0>P<0>P<0>I<0>O<0>M",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 3
      IF (MASK("<0>P<0>P<0>I<0>N<0>A",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 4
      IF (MASK("<0>R<0>O<0>B<0>E<0>P<0>S",6,IUP,IBGN,IEND).NE.-1)
     +        GOTO 100
      ITYP = 5
      IF (MASK("<0>A<0>R<0>N<0>O",4,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 6
      IF (MASK("<0>R<0>H<0>I<0>N<0>O",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 0
  100 RETURN
      END
```

```
      FUNCTION MASK(MSK,LMSK,IUP,IBGN,ISTOP)
      DIMENSION MSK(1), IUP(1)
C
C      THIS FUNCTION SEARCHES IUP (AN UNPACKED ARRAY OF ASCII CHARACTERS) FROM
C      IBGN TO ISTOP FOR A STRING WHICH MATCHES THE MASK MSK.  MSK IS LMSK
C      WORDS IN LENGTH AND HAS THE FOLLOWING CHARACTERISTICS:
C
C           1.  IF THE LEFT BYTE OF A WORD IS 0, THE RIGHT BYTE NEEDS AN
C               EXACT MATCH.
C           2.  IF THE LEFT BYTE IS NON-ZERO, THEN A MATCH MUST BE GREATER
C               THAN OR EQUAL TO THE LEFT BYTE, AND LESS THEN OR EQUAL TO THE
C               RIGHT BYTE.  "09" OR "AZ", E. G.
C           3.  IF A WORD EQUALS -1, IT WILL MATCH ANY BYTE (A WILD CARD).
C
C      THE FUNCTION RETURNS THE LOCATION IN IUP OF THE BEGINNING OF THE
C      SUBSTRING.  MASK RETURNS -1 IF THE SEARCH FAILS.
C
      LIMIT = ISTOP - LMSK + 1
      DO 600 MASK = IBGN, LIMIT
       DO 500 I = 1, LMSK
        M = MSK(I)
        IF (M.EQ.-1) GOTO 500              ;WILD CARD.
        ML = ISHFT(M,-8)
        MR = IAND(M,377K)
        L = IUP(MASK+I-1)
        IF (ML.NE.0) GOTO 200              ;NEED AN EXACT MATCH.
         IF (MR.EQ.L) GOTO 500
         GOTO 600
200     IF (L.LT.ML.OR.L.GT.MR) GOTO 600   ;RANGE FOR MATCH.
500    CONTINUE
       GOTO 700
600   CONTINUE
      MASK = -1                            ;UNSUCCESSFUL SEARCH.
700   RETURN
      END
```

```
      SUBROUTINE MDRGRID(MDR,IMAX,JMAX,IHR,IAR,IOFF,JOFF)
      DIMENSION MDR(30,30,7), IAR(13,13)
C
C      THIS SUBROUTINE TAKES THE 13X13 STATION GRID (IAR) AND FITS IT INTO
C      THE PROPER HOUR OF THE REGIONAL GRID (MDR).  IOFF AND JOFF ARE ADDED
C      TO THE IAR COORDINATES TO GET COORDINATES IN MDR.  THE HIGHEST VALUE
C      IS TAKEN EXCEPT:
C
C           1.  SEVEN IS ALWAYS REPLACED.  (MISSING DATA.)
C           2.  ONE THROUGH SIX REPLACE EIGHT OR NINE.  (UNK INTENSITY.)
C           3.  EIGHT ONLY REPLACES ZERO, SEVEN, AND NINE.  (UNK, BUT SVR.)
C           4.  NINE ONLY REPLACES ZERO AND SEVEN.
C
      DO 500 I = 1, 13                              ;LOOP FOR EACH IAR POINT.
       DO 500 J = 1, 13
        IG = I + IOFF                               ;MAKE SURE THIS POINT IS ON
        IF (IG.LE.0) GOTO 500                       ;THE REGIONAL GRID.
        IF (IG.GT.IMAX) GOTO 500
        JG = J + JOFF
        IF (JG.LE.0) GOTO 500
        IF (JG.GT.JMAX) GOTO 500
        NEW = IAR(I,J)                              ;GET VALUES FROM ARRAYS.
        IOLD = MDR(IG,JG,IHR)
        IF (IOLD.EQ.7) GOTO 400                     ;SEVEN IS ALWAYS REPLACED.
        IF (NEW.EQ.0) GOTO 500                      ;ZERO ONLY REPLACES SEVEN.
        IF (NEW.NE.8) GOTO 100                      ;EIGHT ONLY REPLACES ZERO,
         IF (IOLD.EQ.0) GOTO 400                    ;SEVEN, OR NINE.
         IF (IOLD.EQ.9) GOTO 400
         GOTO 500
100     IF (NEW.NE.9) GOTO 200                      ;NINE ONLY REPLACES ZERO OR
         IF (IOLD.EQ.0) GOTO 400                    ;SEVEN.
         GOTO 500
200     IF (IOLD.NE.8) GOTO 300                     ;ONE-SIX REPLACE EIGHT.
         GOTO 400
300     IF (IOLD.EQ.9) GOTO 400                     ;ONE-SIX REPLACE NINE.
        IF (IOLD.GE.NEW) GOTO 500                   ;TAKE THE LARGEST.
400     MDR(IG,JG,IHR) = NEW
500   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MDRCND(MDR, IMAX,JMAX)
      DIMENSION MDR(30,30,7)
C
C      THIS SUBROUTINE "CONDITIONS" THE MDR GRID.  IF THERE IS NO DATA FOR A
C      GRID BOX, THE VALUES ARE CHECKED FOR THE SAME BOX ONE HOUR BEFORE AND
C      ONE HOUR AFTER.  AN AVERAGE VALUE IS INSERTED IF POSSIBLE.
C
      DO 300 I = 1, IMAX
       DO 300 J = 1, JMAX
        DO 300 K = 1, 7
         IF (MDR(I,J,K).NE.7) GOTO 300
         IF (K.NE.1) GOTO 100                        ;1ST HOUR MSG; USE 2ND.
          MDR(I,J,1) = MDR(I,J,2)
          GOTO 300
100      IF (K.NE.7) GOTO 200                        ;7TH HOUR MSG; USE 6TH.
          MDR(I,J,7) = MDR(I,J,6)
          GOTO 300
200      IF (MDR(I,J,K-1).GE.7) GOTO 300             ;AVERAGE THE TWO ON
         IF (MDR(I,J,K+1).GE.7) GOTO 300             ;EITHER SIDE.
         MDR(I,J,K) = (MDR(I,J,K-1)*10 + MDR(I,J,K+1)*10 + 5)/20
300   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RR(MDR,IRR,IRFL,IMAX,JMAX)
      DIMENSION MDR(30,30,7), IRFL(30,30,3)
      COMMON /Z2R/ IV2R(10,3)
C
C      THIS SUBROUTINE USES THE CONDITIONED MDR GRID TO PRODUCE A RAINFALL
C      GRID (IRFL).  IRR DETERMINES WHICH RELATION IN IV2R WILL BE USED.
C
C      IRFL(I,J,K) RAINFALL GRID.  THE THREE HIGHEST-ORDER BITS OF EACH VALUE
C      ARE RESERVED FOR INCOMPLETE DATA (1B0), LEVEL 5 OR 6 (1B1), AND FUTURE
C      EXPANSION.  THE REMAINDER OF EACH VALUE SPECIFIES RAINFALL IN HUNDRETHS
C      OF AN INCH.
C         I:  I-COORD OF GRID BOX.
C         J:  J-COORD OF GRID BOX.
C         K:  1 = CURRENT RAFL RATE.  2 = 3-HR ACCUM.  3 = 6-HR ACCUM.
C      IV2R(I,J) VIP LEVEL TO RAINFALL CONVERSION.
C         I:  VIP LEVEL + 1.  (J = 1 FOR VIP LEVEL 0).
C         J:  Z-R RELATION.  1 = CONVECTV.  2 = STRATOFRM.  3 = USER-DEFINED.
C
      DO 500 I = 1, IMAX                              ;LOOP FOR EACH GRID POINT.
      DO 500 J = 1, JMAX
      IR = 0                                          ;DEFAULT VALUES.
      MSG = 0
      I56 = 0
      M1 = MDR(I,J,1)                                 ;FREQUENTLY-USED ARRAY
      M2 = MDR(I,J,2)                                 ;VALUES.
      M3 = MDR(I,J,3)
      M4 = MDR(I,J,4)
      M5 = MDR(I,J,5)
      M6 = MDR(I,J,6)
      M7 = MDR(I,J,7)
      IF (M1.EQ.5.OR.M2.EQ.5.OR.M1.EQ.6.OR.M2.EQ.6) I56 = 1    ;COMPUTE 1-HR
      IF (M1.NE.7.AND.M1.NE.8.AND.M1.NE.9) GOTO 100            ;RAINFALL.
       MSG = 1
       IF (M2.NE.7.AND.M2.NE.8.AND.M2.NE.9) IR = IV2R(M2+1,IRR)
       GOTO 200
  100 IF (M2.NE.7.AND.M2.NE.8.AND.M2.NE.9) GOTO 150
       MSG = 1
       IR = IV2R(M1+1,IRR)
       GOTO 200
  150 IR = ((IV2R(M1+1,IRR) + IV2R(M2+1,IRR))*10 + 5)/20
  200 IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,1) = IOR(IR1,ISHFT(I56,14))
      IF (M3.EQ.5.OR.M4.EQ.5.OR.M3.EQ.6.OR.M4.EQ.6) I56 = 1    ;COMPUTE 3-HR
      IF (M3.EQ.7.OR.M3.EQ.8.OR.M3.EQ.9) MSG = 1               ;RAINFALL.
      IF (M4.EQ.7.OR.M4.EQ.8.OR.M4.EQ.9) MSG = 1
      IR = ((IV2R(M1+1,IRR) + IV2R(M4+1,IRR))*10 + 5)/20 +
     &         IV2R(M2+1,IRR) + IV2R(M3+1,IRR)
      IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,2) = IOR(IR1,ISHFT(I56,14))
      IF (M5.EQ.5.OR.M6.EQ.5.OR.M7.EQ.5) I56 = 1   ;COMPUTE 6-HR RAINFALL.
      IF (M5.EQ.6.OR.M6.EQ.6.OR.M7.EQ.6) I56 = 1
      IF (M5.EQ.7.OR.M6.EQ.7.OR.M7.EQ.7) MSG = 1
      IF (M5.EQ.8.OR.M6.EQ.8.OR.M7.EQ.8) MSG = 1
      IF (M5.EQ.9.OR.M6.EQ.9.OR.M7.EQ.9) MSG = 1
      IR = IR + ((IV2R(M4+1,IRR) + IV2R(M7+1,IRR))*10 + 5)/20 +
     &              IV2R(M5+1,IRR) + IV2R(M6+1,IRR)
      IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,3) = IOR(IR1,ISHFT(I56,14))
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE RRM(MDR,IRFL,IMAX,JMAX)
      DIMENSION MDR(30,30,7), IRFL(30,30,3)
C
C       THIS SUBROUTINE TOTALS MDR NUMBERS.  ARRAYS AND FORMATS ARE THE SAME
C       AS RR.FR.
C
      DO 500 I = 1, IMAX                                   ;LOOP FOR EACH GRID POINT.
      DO 500 J = 1, JMAX
                                                           ;SET UP DEFAULTS.
      IR = 0
      MSG = 0
      I56 = 0
                                                           ;FREQUENTLY USED VALUES.
      M1 = MDR(I,J,1)
      M2 = MDR(I,J,2)
      M3 = MDR(I,J,3)
      M4 = MDR(I,J,4)
      M5 = MDR(I,J,5)
      M6 = MDR(I,J,6)
                                                           ;1-HR MDR.
      IF (M1.EQ.5.OR.M1.EQ.6) I56 = 1
      IF (M1.EQ.7.OR.M1.EQ.8.OR.M1.EQ.9) MSG = 1
      IF (M1.LE.6) IR = M1*10
      IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,1) = IOR(IR1,ISHFT(I56,14))
                                                           ;3-HR MDR.
      IF (M2.EQ.5.OR.M3.EQ.5) I56 = 1
      IF (M2.EQ.6.OR.M3.EQ.6) I56 = 1
      IF (M2.EQ.7.OR.M3.EQ.7) MSG = 1
      IF (M2.EQ.8.OR.M3.EQ.8) MSG = 1
      IF (M2.EQ.9.OR.M3.EQ.9) MSG = 1
      IF (M2.LE.6) IR = IR + M2*10
      IF (M3.LE.6) IR = IR + M3*10
      IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,2) = IOR(IR,ISHFT(I56,14))
      IF (M4.EQ.5.OR.M5.EQ.5.OR.M6.EQ.5) I56 = 1          ;6-HR MDR TOTALS.
      IF (M4.EQ.6.OR.M5.EQ.6.OR.M6.EQ.6) I56 = 1
      IF (M4.EQ.7.OR.M5.EQ.7.OR.M6.EQ.7) MSG = 1
      IF (M4.EQ.8.OR.M5.EQ.8.OR.M6.EQ.8) MSG = 1
      IF (M4.EQ.9.OR.M5.EQ.9.OR.M6.EQ.9) MSG = 1
      IF (M4.LE.6) IR = IR + M4*10
      IF (M5.LE.6) IR = IR + M5*10
      IF (M6.LE.6) IR = IR + M6*10
      IR1 = IOR(IR,ISHFT(MSG,15))
      IRFL(I,J,3) = IOR(IR,ISHFT(I56,14))
  500 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MDRGPH(IDTMS,IBF,IRR,IRFL,IMAX,JMAX)
      DIMENSION IBF(0:255), IRFL(30,30,3), LINE(2), IDTMS(4,7)
      LOGICAL ITEST
C
C      THIS SUBROUTINE USES THE IRFL ARRAY TO GENERATE THREE GRAPHIC PRODUCTS:
C                    RR1, INSTANTANEOUS RAINFALL RATE,
C                    RR3, THREE-HOUR RAINFALL ACCUMULATION, AND
C                    RR6, SIX-HOUR RAINFALL ACCUMULATION.
C
C      THE STATION MODEL LOOKS LIKE THIS:
C                                              THE TWO DIGITS ARE INCHES
C                                              AND TENTHS OF RAINFALL.
C      THE M APPEARS IF DATA ARE      15
C      ARE MISSING FOR THIS POINT.    M*
C                                              THE ASTERISK APPEARS IF A LEVEL
C                                              5 OR 6 OCCURED AT THIS POINT.
C
C      THE LOWER LEFT CORNER OF THE FIRST DIGIT IS CENTERED OVER THE GRID BOX.
C
       IUL = IBF(160)                                ;CONVERTS REGIONAL TO
       JUL = IBF(161)                                ;NATIONAL GRID.
       MAP = IBF(172)                                ;GET MAP NUMBER.
       LINE(2) = 0
       CALL MDRCOORD(IUL+IMAX+1,JUL,XLAT,XLON)
       CALL CART(XLAT,XLON,MAP,LX,LY,IER)
       DO 600 K = 1, 3                               ;LOOP FOR GRAPHICS.
       DO 500 I = 1, IMAX                            ;LOOP FOR GRID BOXES.
       DO 500 J = 1, JMAX
       CALL MDRCOORD(IUL+I-1,JUL+J-1,XLAT,XLON)      ;GET LAT AND LON.
       CALL CART(XLAT,XLON,MAP,IX,IY,IER)            ;GET GRAPHIC PIXELS.
       L = IRFL(I,J,K)
       M = 20040K                                    ;MISSING.
       IF (ITEST(L,15)) M = 46440K
       IF (ITEST(L,14)) M = IOR(IAND(M,177400K),52K)  ;LEVELS 5 OR 6.
       LINE(1) = M
       CALL TEXT(LINE,IX,IY,0,1,-1,-10)
       L = IAND(L,17777K) + 5                        ;ROUND-OFF FOR PRECIP.
       M = 20040K                                    ;CONVERT PRECIP. TO ASCII.
       IF (L.GE.100) M = IASCL(L,100)
       LINE(1) = IOR(M,IASCR(L,10))
       CALL TEXT(LINE,IX,IY,0,1,-1,-1)
   500 CONTINUE
       CALL MDRLB(K,IRR,IDTMS,LX,LY)                 ;SET UP LABELS.
   600 CONTINUE
       RETURN
       END
```

```
      SUBROUTINE MDRLB(IGPH,IRR,IDTMS,LX,LY)
      DIMENSION IDTMS(4,7)
      COMMON /LBLS/ LABEL(18)
      DATA LABEL /"1-HR INST CONV RAFL 00/00/00 0035Z<0>"/
C
C       THIS SUBROUTINE GENERATES THE LABELS FOR THE THREE GRAPHIC PRODUCTS.
C       THEN IT CLOSES THE FILES AND SENDS THEM TO THE DATABASE.
C
      GOTO (110,120,130,140),IRR                          ;TEST FOR RAFL RELATION.
  110 GOTO 200                                            ;CONVECTIVE.
  120 LABEL(6) = "ST"                                     ;STRATOFORM.
      LABEL(7) = "RT"
      GOTO 200
  130 LABEL(6) = "OP"                                     ;OPTIONAL.
      LABEL(7) = "TN"
      GOTO 200
  140 LABEL(3) = " M"                                     ;MDR TOTALS.
      LABEL(4) = "DR"
      LABEL(5) = " T"
      LABEL(6) = "OT"
      LABEL(7) = "AL"
      LABEL(8) = "S "
      LABEL(9) = "  "
      LABEL(10) = "  "
  200 GOTO (210,220,230),IGPH                             ;CHECK GRAPHIC.
  210 GOTO 300                                            ;1-HR GRAPHIC.
  220 LABEL(1) = "3-"                                     ;3-HR GRAPHIC.
      IF (IRR.EQ.4) GOTO 300
      LABEL(3) = " A"
      LABEL(4) = "CC"
      LABEL(5) = "M "
      GOTO 300
  230 LABEL(1) = "6-"
  300 LABEL(11) = IOR(IASCL(IDTMS(1,1),10),IASCR(IDTMS(1,1),1))     ;LOAD DATE
      LABEL(12) = IOR(27400K,IASCR(IDTMS(2,1),10))                  ;AND TIME.
      LABEL(13) = IOR(IASCL(IDTMS(2,1),1),57K)
      LABEL(14) = IOR(IASCL(IDTMS(3,1),10),IASCR(IDTMS(3,1),1))
      LABEL(15) = IOR(20000K,IASCR(IDTMS(4,1),10))
      LABEL(16) = IOR(IASCL(IDTMS(4,1),1),63K)
      CALL TEXT(LABEL,LX,LY,1,1,0,0)                      ;WRITE THE LABEL.
      GOTO (410,420,430),IGPH                             ;STORE THE GRAPHICS.
  410 CALL UTF("NMCGPHRR1","RR1")                         ;1-HR.
      GOTO 500
  420 CALL UTF("NMCGPHRR3","RR3")                         ;3-HR.
      GOTO 500
  430 CALL UTF("NMCGPHRR6","RR6")                         ;6-HR.
  500 RETURN
      END
```

```
      SUBROUTINE MDRCOORD(I,J,XLAT,XLON)
C
C       THIS SUBROUTINE TAKES THE (I,J) COORDINATES OF A POINT ON THE NATIONAL
C       MDR GRID AND PRODUCES A LATITUDE AND LONGITUDE.
C
      XJ = -FLOAT(I) - 70.5                  ;CONVERT TO LFM GRID.
      XI = FLOAT(J) - 41.5
      R2 = XI*XI + XJ*XJ
      XLAT = 57.2957795*ASIN((62317.4599-R2)/(62317.4599+R2))    ;GET LAT/LON.
      X = 57.2957795*ATAN2(XJ,XI)
      IF (X) 10,20,20                        ;CONVERT BACK FROM LFM GRID.
   10 XLON = 15. - X
      GOTO 30
   20 XLON = 385. - X
   30 RETURN
      END
```

```
      REAL FUNCTION ASIN(X)
C
C      THIS FUNCTION USES A MCLAURIN SERIES TO COMPUTE THE ARC SIN OF THE
C      INPUT VARIABLE X.  OUTPUT IS IN RADIANS.
C
      AS = X
      TERM = X
      X2 = X*X
      DO 100 I = 1,2001,2
      FCTR = FLOAT(I)*FLOAT(I)/(FLOAT(I+1)*FLOAT(I+2))
      TERM = TERM*X2*FCTR
      IF (TERM.LE.0.0000001) GOTO 200
      AS = AS + TERM
  100 CONTINUE
  200 ASIN = AS
      RETURN
      END
```

```
      INTEGER FUNCTION JDATE(MONTH,IDAY,IYEAR)
C
C      THIS FUNCTION WILL GENERATE A JULIAN DATE FOR THE MONTH, DAY, AND
C      YEAR INPUT.  IF THERE IS A PROBLEM WITH THE INPUT DATA, THE FUNC-
C      TION WILL RETURN A VALUE OF ZERO.
C
      COMMON /QJULQ/ MLIST(12)
      DATA MLIST/31,28,31,30,31,30,31,31,30,31,30,31/
      JDATE = 0                           ;CHECK INPUT DATA.
      IF (MONTH.LT.1.OR.MONTH.GT.12) RETURN
      IF (IYEAR.LT.1) RETURN
      IF (IDAY.LT.1) RETURN
      MLIST(2) = 28
      I = IYEAR/4*4                       ;IS THIS A LEAP YEAR?
      IF (I.EQ.IYEAR) MLIST(2) = 29
      I = IYEAR/100*100
      IF (I.EQ.IYEAR) MLIST(2) = 28
      I = IYEAR/400*400
      IF (I.EQ.IYEAR) MLIST(2) = 29
      IF (IDAY.GT.MLIST(MONTH)) RETURN
      DO 100 I = 1,MONTH                  ;COMPUTE JULIAN DATE.
  100 JDATE = JDATE + MLIST(I)
      JDATE = JDATE - MLIST(MONTH) + IDAY
      RETURN
      END
```

```
      SUBROUTINE J2MDA(JDA,M,ID,IY)
      DIMENSION MS(12)
C
C       THIS SUBROUTINE ACCEPTS A JULIAN DATE (JDA) AND A YEAR (IY).  IT
C       PRODUCES A MONTH (M) AND A DAY (ID).  IF THE JULIAN DATE IS BAD,
C       M AND ID WILL BOTH EQUAL ZERO.
C
      M = 0
      ID = 0
      IF (JDA.LE.0) RETURN                       ;CHECK FOR BAD JDA.
      MS(1) = 31                                 ;SET UP MONTH DATA.
      MS(2) = 28
      I = IY/4*4                                 ;LEAP YEARS.
      IF (I.EQ.IY) MS(2) = 29
      I = IY/100*100
      IF (I.EQ.IY) MS(2) = 28
      I = IY/400*400
      IF (I.EQ.IY) MS(2) = 29
      MS(3) = 31
      MS(4) = 30
      MS(5) = 31
      MS(6) = 30
      MS(7) = 31
      MS(8) = 31
      MS(9) = 30
      MS(10) = 31
      MS(11) = 30
      MS(12) = 31
      J = 0                                      ;SEARCH FOR MONTH.
      DO 100 I = 1,12
      M = I
      J = J + MS(I)
      IF (J.GE.JDA) GOTO 200
  100 CONTINUE
      M = 0                                      ;JDA IS TOO BIG.
      RETURN
  200 IF (J.GT.JDA) GOTO 300
      ID = MS(M)
      RETURN
  300 ID = JDA + MS(M) - J
      RETURN
      END
```

```
      INTEGER FUNCTION IASCR(NUM,IPR)
C
C      THIS FUNCTION TAKES THE IPR'TH DECIMAL PLACE NUM, AND CONVERTS
C      IT TO AN ASCII CHARACTER WHICH IS RETURNED IN THE RIGHT BYTE
C      OF THE RETURN.
C
      IASCR = (NUM - (NUM/(10*IPR))*(10*IPR))/IPR + 48
      RETURN
      END
```

```
      INTEGER FUNCTION IASCL(NUM,IPR)
C
C      THIS FUNCTION TAKES THE IPR'TH DECIMAL PLACE NUM, AND CONVERTS
C      IT TO AN ASCII CHARACTER WHICH IS RETURNED IN THE LEFT BYTE
C      OF THE RETURN.
C
      IASCL = ISHFT(((NUM - (NUM/(10*IPR))*(10*IPR))/IPR + 48),8)
      RETURN
      END
```

```
DELETE DP0F:MDR.SV
RLDR MDR MDR<UN SU> ROBDEC GROB ROBOP MASK MDR<GRID CND> RR RRM^
 MDRGPH MDRLB MDRCOORD ASIN JDATE J2MDA IASC<R L>^
 <AG AG TOP BG UTIL FORT>.LB MDR.LM/L
CHATR DP0F:MDR.SV S
```

```
NNNNC                 MDRGEN.SV           SETS UP THE MDRIT FILE.
      COMMON /MDRDT/ IBF(0:255)
      DATA IBF /256*0/
C
C       THIS PROGRAM SETS UP A FILE NAMED MDRIT (MDR INFORMATION TABLE).
C       THIS FILE SHOWS WHICH DATABASE PRODUCTS MUST BE DECODED AND THE
C       COORDINATES AND SIZE OF THE REGION TO BE MONITORED.
C
C                 RLDR MDRGEN MDRG1 MDRG2 MDRG3 UTIL.LB FORT.LB
C
      CALL MDRG1(IBF)                                 ;LOAD STATION DATA.
      CALL MDRG2(IBF)                                 ;LOAD REGIONAL DATA.
      CALL MDRG3(IBF)                                 ;LOAD RAINFALL DATA.
      CALL CRAND("MDRIT",IER)
      IF (IER.EQ.1) GOTO 300
       WRITE (10,900)                                 ;FILE ALREADY EXISTS!
       READ  (11,910) IANS
       IF (ISHFT(IANS,-8).EQ.131K) GOTO 200
       TYPE "NEW DATA LOST.  BETTER LUCK NEXT TIME."
       GOTO 400
  200  CALL DFILW("MDRIT",IER)
       CALL ERROR(IER,"CAN'T DELETE MDRIT--")
       CALL CRAND("MDRIT",IER)
       TYPE "OLD MDRIT DELETED."
  300 CALL GCHN(IC,IER)
      CALL OPENE(IC,"MDRIT",0,IER)
      CALL WRB(IC,IBF,0,1,IER)
      CALL KLOSE(IC,IER)
      TYPE "NEW MDRIT FILE CREATED.  HOPE IT WORKS."
  400 CALL EXIT
  900 FORMAT(" MDRIT ALREADY EXISTS.  DO YOU WANT TO REPLACE",
     +" IT (Y/N)?      ",Z)
  910 FORMAT(A2)
      END
```

```
      SUBROUTINE MDRG1(IBF)
      DIMENSION IBF(0:255), IPROD(5)
C
C      THIS SUBROUTINE LOADS THE DATA FOR ALL THE INDIVIDUAL RADAR SITES
C      WHICH MUST BE LOADED INTO MDRIT.
C
      DO 200 I = 1, 20
      J = (I-1) * 8
      WRITE (10,900) I                                  ;GET CCCNNNXXX.
      READ  (11,910) IPROD
      IF (IPROD(1).EQ.20040K) RETURN
      IPROD(5) = IAND(IPROD(5),177400K)                 ;ZERO OUT TENTH BYTE.
      DO 100 K = 1, 5
  100 IBF(J+K-1) = IPROD(K)
      WRITE (10,920)                                    ;NETWORK RADAR?
      READ  (11,930) IANS
      IF (ISHFT(IANS,-8).EQ.131K) IBF(J+5) = 1
      J6 = J + 6
      J7 = J + 7
      ACCEPT "RADAR COORDINATES (I,J)?      ",IBF(J6),IBF(J7)
  200 CONTINUE
      TYPE "TWENTY STATIONS IS THE MAXIMUM."
      RETURN
  900 FORMAT(" CCCNNNXXX FOR STATION",I3,"?  (HIT RETURN TO MOVE ON.)",
     +5X,Z)
  910 FORMAT(4A2,A1)
  920 FORMAT(" IS THIS A NETWORK RADAR (Y/N)?      ",Z)
  930 FORMAT(A2)
      END
```

```
      SUBROUTINE MDRG2(IBF)
      DIMENSION IBF(0:255)
C
C       THIS SUBROUTINE DETERMINES THE REGIONAL COORDINATES AND SIZE.  IT THEN
C       CHANGES THE INDIVDUAL STATION COORDINATES INTO THE NECESSARY OFFSETS.
C
  100 ACCEPT
     +"COORDS OF THE UPPER LEFT CORNER OF THE REGION (I,J)?        ",
     +I160, I161
      IBF(160) = I160
      IBF(161) = I161
      ACCEPT "COORDS OF THE LOWER RIGHT CORNER (I,J)?        ", ILR, JLR
      IBF(162) = ILR - I160 + 1                     ;COMPUTE HEIGHT OF REGION.
      IBF(163) = JLR - I161 + 1                     ;COMPUTE WIDTH OF REGION.
      IF (IBF(162).LE.30.AND.IBF(163).LE.30) GOTO 200
      TYPE "REGION IS TOO LARGE--30 X 30 IS MAXIMUM."
      GOTO 100                                      ;NOW CONVERT STATION COORDS
  200 DO 300 I = 1, 20                              ;INTO OFFSETS.
       J = (I-1) * 8
       IF (IBF(J).EQ.0) RETURN
       J6 = J + 6
       J7 = J + 7
       IBF(J6) = IBF(J6) - I160 - 6
       IBF(J7) = IBF(J7) - I161 - 6
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MDRG3(IBF)
      DIMENSION IBF(0:255)
C
C      THIS SUBROUTINE LOADS THE OPTIONAL RAINFALL RELATION INTO MDRIT AND
C      SETS THE MAP BACKGROUND AND DEFAULT RAINFALL RELATION OPTIONS.
C
      WRITE (10,900)                                ;DECIDE WHETHER TO LOAD IT.
      READ  (11,910) I
      IF (ISHFT(I,-8).NE.131K) GOTO 200
      DO 100 I = 1, 6                               ;VIP LEVELS 1 - 6.
       J = 163 + I
       WRITE (10,920) I
       ACCEPT IBF(J)
  100 CONTINUE
      TYPE "VIP 8 INDICATES UNKNOWN INTENSITY, BUT SEVERE."
      ACCEPT "RAINFALL (1/100'S) FOR VIP 8?      ", IBF(170)
      TYPE "VIP 9 INDICATES UNKNOWN INTENSITY."
      ACCEPT "RAINFALL (1/100'S) FOR VIP 9?      ", IBF(171)
  200 ACCEPT "MAP BACKGROUND?      ",IBF(172)
      TYPE "DEFAULT RAINFALL RELATION"
      TYPE "    (1 = CONVECTIVE   2 = STRATOFORM"
      ACCEPT "     3 = OPTIONAL      4 = MDR TOTALS)?      ",IBF(173)
      RETURN
  900 FORMAT(" DO YOU WANT TO DEFINE AN OPTIONAL RAINFALL RELATION",
     +" (Y/N)?      ",Z)
  910 FORMAT(A2)
  920 FORMAT(" RAINFALL (1/100'S) FOR VIP ",I1,"?      ",Z)
      END
```

APPENDIX I

In deriving rainfall totals from MDR observations, the MDR grid square is treated as a rain gage, 22n.m. on a side. The derived rainfall will represent an average across the box but will seldom be observed at a single point.

Deriving rainfall from radar observations is fraught with problems and the problems are compounded when the radar observations are converted to MDR code. The MDR code characterizes each grid square by the highest VIP level observed within it. In reality, high VIP levels cover only a small fraction of the grid square. Instantaneous MDR observations are available once per hour. The rainfall for any hour is computed from only two MDR observations --- at the beginning and end of the hour. Anyone who has viewed a precipitation event on radar knows that frequent changes can take place in one hour. Therefore, inadequate temporal and spatial resolution are among the problems faced in deriving rainfall from radar observations.

Attempts were made to increase, both the temporal and spatial resolution of MDR observations using a typical areal coverage of various VIP levels (Brandes 1973). A comparison of rainfall derived from MDR observations with rain gage rainfall led to the conclusion that the refinement of spatial resolution significantly improved the MDR-derived rainfall. No attempt was made to refine the temporal resolution after it was determined that it would not have the same impact on derived rainfall as did the improved spatial resolution. The following assumptions were used in refining the spatial resolution:

1. The maximum VIP level is centered in the MDR grid square.

2. The radar cells are circular.

3. The various VIP levels are concentric circles within the cell.

The typical areal coverages of each reflectivity level, determined by Brandes (1973), were adjusted for the MDR observational system. Each areal coverage was divided by the area of an MDR grid square (484n.m.) to arrive at the fractional coverage of the grid square by each VIP level. Subtracting the fractional coverage of the next higher VIP level from the fractional coverage of the VIP level being processed and multiplying by the rainfall rate of the VIP level being processed gives the rainfall contribution of that VIP level. Table 3 provides the rainfall rate, fractional coverage, and rainfall rate adjusted for fractional coverage for each VIP level. An example of an adjusted rainfall rate calculation is also provided.

In the accompanying software, the convective rainfall rate is modified for fractional coverage of the MDR grid square and the stratiform rainfall rate is unmodified. An optional rainfall rate is described in Table 3 which is the stratiform rainfall rate modified for fractional coverage. Modification of the stratiform rainfall rate is questionable since Brandes (1973) derived his results for convective events. In numerous comparisons of MDR derived rainfall to raingage observations, the following has been found:

1. In convective events, MDR derived rainfall consistently overestimates raingage rainfall by a factor of about 2.

2. In stratiform events, MDR rainfall, from the stratiform rainfall rate, overestimates raingage rainfall by a factor of 2-7 --- especially in areas of maximum precipitation. This may be due to VIP level 3 cells embedded within a general area of VIP level 1 and 2 cells.

3. A side-by-side comparison of the MDR rainfall for the optional rainfall rate, in the events of 2., above, shows an overestimate of raingage rainfall by a factor of about 2 in areas of maximum precipitation. In other areas, the optional rainfall rate tends to slightly underestimate rainfall.

It is left to the user to determine the proper rainfall rate for stratiform events. Since the relatively gross nature of MDR observations may preclude any further refinement of the stratiform rainfall rate, it is recommended that the stratiform and optional rainfall rates be run side-by-side until the user can identify events in which one will outperform the other.

TABLE 3

RADAR REFLECTIVITY/RAINFALL RATE (Z/R)
UNMODIFIED AND MODIFIED FOR FRACTIONAL
COVERAGE OF MDR GRID SQUARE

VIP	Fractional Coverage of MDR Grid Square	Unmodified Convective * Rainfall Rate	Modified Convective Rainfall Rate	Unmodified Stratiform * Rainfall Rate	Modified Stratiform Rainfall Rate
1	1.00	0.10 in/hr	0.10 in/hr	0.05 in/hr	0.05 in/hr
2	0.68	0.65 "	0.47 "	0.30 "	0.22 "
3	0.16	1.65 "	0.63 "	0.75 "	0.29 "
4#	0.079	3.35 "	0.77 "	0.77 "	0.77 "
5#	0.031	5.80 "	0.84 "	0.84 "	0.84 "
6#	0.002	7.10 "	0.85 "	0.85 "	0.85 "

* rainfall rates are the midpoint of the range given in NWS Radar Code User's Guide (1980).

\# no rainfall rate is given for stratiform events for VIP 4 and greater; therefore the modified convective rate is used.

Example of calculation of modified convective rainfall rate for MDR6:

$$\hat{R}6 = (0.002 \times R6) + (0.029 \times R5) + (0.048 \times R4) + (0.081 \times R3) + (0.52 \times R2) + (0.32 \times R1) =$$

$$(0.002 \times 7.10) + (0.029 \times 5.80) + (0.048 \times 3.35) + (0.081 \times 1.65) + (0.52 \times 0.65) + (0.32 \times 0.10) = 0.85$$

where Rn = unmodified convective rainfall rate in above table for VIP=n

```
MDRGEN
CCCNNNXXX FOR STATION  1?  (HIT RETURN TO MOVE ON.)      CLEROBCLE
IS THIS A NETWORK RADAR (Y/N)?    N
RADAR COORDINATES (I,J)?     33,86
CCCNNNXXX FOR STATION  2?  (HIT RETURN TO MOVE ON.)      CLEROBCVG
IS THIS A NETWORK RADAR (Y/N)?    Y
RADAR COORDINATES (I,J)?     41,83
CCCNNNXXX FOR STATION  3?  (HIT RETURN TO MOVE ON.)      ARBROBCTW
IS THIS A NETWORK RADAR (Y/N)?    Y
RADAR COORDINATES (I,J)?     32,82
CCCNNNXXX FOR STATION  4?  (HIT RETURN TO MOVE ON.)
COORDS OF THE UPPER LEFT CORNER OF THE REGION (I,J)?    30,79
COORDS OF THE LOWER RIGHT CORNER (I,J)?    42,90
DO YOU WANT TO DEFINE AN OPTIONAL RAINFALL RELATION (Y/N)?    N
MAP BACKGROUND?    2
DEFAULT RAINFALL RELATION
   (1 = CONVECTIVE   2 = STRATOFORM
    3 = OPTIONAL     4 = MDR TOTALS)?    1
NEW MDRIT FILE CREATED.  HOPE IT WORKS.
R
```

Figure 4. **A Sample run of MDRGEN.SV**
Operator responses are underlined.

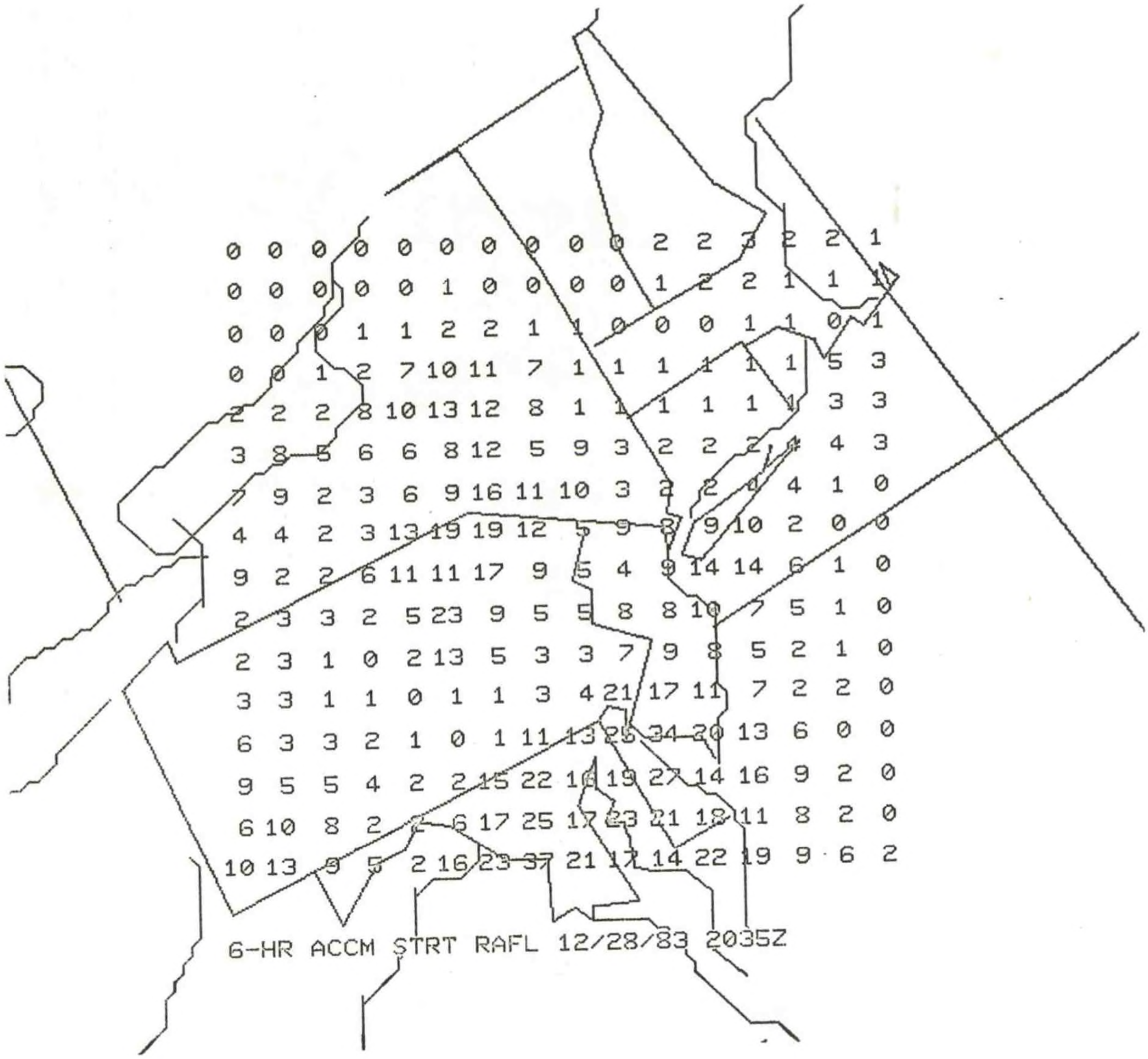

Figure 5. Six accumulated rainfall ending at 2035Z, 12/28/83 based on stratiform rainfall rate. Plotted values are units and tenths of an inch.

EASTERN REGION CP No. 15

MDR--PROCESSING MDR OBSERVATIONS

PART A: INFORMATION AND INSTALLATION

PROGRAM NAME: MDR.SV

AAL ID:
REVISION NO: 1.00

PURPOSE: MDR reads seven hours of radar observations from the local database and produces three graphics. The graphics are one-, three-, and six-hourly sums of MDR values or rainfall amounts. Convective and stratiform Z-R relations are provided along with the option for a user-defined relation.

PROGRAM INFORMATION:

Development Programmer:
Matthew R. Peroutka
Location: WSFO Cleveland, OH
Phone: (FTS) 942-4949
Language: FORTRAN
Date: 11/10/83
Running Time: 1-2 minutes

Maintenance Programmer:
Same

Type: Normal
Revision Date:

Disk Space: Program Files - 54 RDOS blocks
Data Files - 1 RDOS block

PROGRAM REQUIREMENTS

Program Files:

Name	Comments
MDR.SV	Actual program
MDRGEN.SV	Needed only for set up

Data Files:

Name	DP Location	R/W	Comments
MDRIT	DPO	Read	Information table
RR1	DPO	Write	Temporary
RR3	DPO	Write	Temporary
RR6	DPO	Write	Temporary

AFOS Products:

ID	Action	Comments
various ROB's	input	Chosen by running MDRGEN. Should be version purged with at least 20 versions to account for special observations.
NMCGPHRR1	output	Map backgrounds should be set to the value chosen when running MDRGEN.
NMCGPHRR3	output	
NMCGPHRR6	output	

LOAD LINE

```
RLDR/P MDR MDR<UN SU> ROBDEC GROB ROBOP MASK MDR<GRID CND> RR RRM
MDR<GPH LB COORD>ASIN JOATE JZMDA JASC<RL> <AGIAG TOP BG UTIL FORT>.LB
```

PROGRAM INSTALLATION

1. Run MDRGEN to generate an MDRIT file in your location. MDRGEN can be run from a floppy as it will only be needed once. MDRGEN asks for a list of ROB's and their coordinates; tables are attached. MDRGEN asks for the coordinates of the region of interest; these can be determined using the attached map. The number of sites and the size of the region affect the run time. If an optional rainfall relation is desired, it is also entered while running MDRGEN.

2. MDR.SV and MDRIT can reside on DPO or DPOF with links to DPO.

EASTERN REGION CP NO. 15

MDR--PROCESSING MDR OBSERVATIONS

PART B: EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: MDR.SV

AAL ID:
REVISION NO: 1.00

PROGRAM EXECUTION

The basic command is RUN:MDR. This will produce MDR graphics for the past one, three, and six hours using the default relation. Global switches control which relation is used.

/C for convective
/M for MDR totals
/O for optional, user-defined
/S for stratiform

Local switches control the time span. Date and time entered are ending date and time for all three graphic products.

/D for day
/H for hour
/M for month

For example: RUN:MDR/S 12/H 10/M 5/D
Use stratiform relation to compute rainfall ending at 12Z on October 5.

ERROR CONDITIONS

If any problems are encountered finding or decoding observations from network radars, an alert is sent to the originating ADM naming the radar site. The program continues, usually without problems.

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

The National Oceanic and Atmospheric Administration was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to assess the socioeconomic impact of natural and technological changes in the environment and to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth.

The major components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

CONTRACT AND GRANT REPORTS — Reports prepared by contractors or grantees under NOAA sponsorship.

ATLAS — Presentation of analyzed data generally in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

TECHNICAL SERVICE PUBLICATIONS — Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

TECHNICAL REPORTS — Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

6009 Executive Boulevard
Rockville, MD 20852