PRL

TECHNICAL NOTE

PRL-TN-86-55

FORTRAN CODE FOR DECODING IPS DATA

By

A.D. Bobra, B.H. Shah and

M.S. Patel

1986

Physical Research Laboratory

Ahmedabad-380009

DOCUMENT CONTROL AND DATA SHEET

1. REPORT NO. : PRL : TN-86-~~PRL~~ 55

2. TITLE AND SUBTITLE :
" FORTRAN CODE FOR DECODING IPS DATA "

3. REPORT DATE : August, 1986
4. TYPE OF REPORT : Technical Note
5. PAGES : 25
6. PRICE : Unpriced
7. NO. OF REFERENCES : 5

8. AUTHORS : A.D. Bobra, B.H. Shah and M.S. Patel

9. a) PURPOSE : DATA PROCESSING
b) USEFUL FOR : DECODING IPS DATA

10. ORIGINATING UNITS/DIVISIONS AND ADDRESS :

Physical Research Laboratory
Navrangpura
Ahmedabad- 380 009

11. SPONSORING AGENCY : Nil

12. ABSTRACT : Data is recorded on the digital magnetic tape at the three field-stations of the IPS project, using a digital data acquisition system. FORTRAN Code described here is used to decode the data & system status, and creates a data file on the Disk.

13. KEYWORDS : Data Processing, FORTRAN CODE

14. DISTRIBUTION STATEMENT : Disttibution Unlimited

15. SECURITY CLASSIFICATION : Unclassified

FORTRAN CODE FOR DECODING IPS DATA

A.D. Bobra, B.H. Shah and M.S. Patel
Physical Research Laboratory
Ahmedabad-380009

Abstract

We present a software program to decode the IPS data, available on digital magnetic tapes from the three IPS field stations, using DEC-10 Systems at PRL. The decoded data is available in a disk file which can be used for further scientific analysis. These programs are written keeping in mind the simplicity, as they are to be used basically to check the IPS system performance. These programs are not by any way the final programs. Lot of improvement could be done to make them more efficient.

Introduction

Large amount of data is being collected at the three field stations (viz. Thaltej, Rajkot and Surat) of the IPS project using digital data acquisition system (DAS). Fig. 1 shows a simple block diagram of the DAS installed at each of the three IPS field stations. The system consists of a IST (Indian Standard Time) and a SDT (Sidereal Time) digital clocks. The IST clocks at each station is synchronized, to National time standard signal (ATA) being transmitted from Delhi, within few msec. The DAS has subsystems like a HEX ENCODER, a SOURCE LIBRARY, a MUX CONTROLLER, Sample and hold (S & H), Analog to Digital

Converter (A/D C), a DATA MULTIPLEXER and a PERTEC INTERFACE. The output of correlation receiver[1] is digitized at an interval of 48 millisecond using S & H and 1 12-bit A/D C. The source library can store information for about 250 radio sources. These information include the SDT ON time, SDT OFF time, beam code, low frequency attenuation etc. The data multiplexer multiplexes the A/D C data and the system status information, and the same is recorded on a magnetic tape using a PERTEC interface and PERTEC digital magnetic tape recorder. Fig. 2 shows the sequence of system-status information as multiplexed by the 1 bit-multiplexer. The information includes the IST from msec to Hour and number of day, the Station Code, Source Code, Beam Code, SDT ON time and the synchronisation word (Sync. Word). The A/D C output (12 bit each for SIN and COS) is encoded with the status bits as shown in Fig. 3. This is the PERTEC format which is in the byte mode. It is important to note that each PERTEC byte has 6 bit of data, 1 bit zero, and 1 bit of status. Thus the entire status bit sequence of 32 digit takes 128 bytes. These cycles are repeated. The record length of the digital magnetic tape is 1024 bytes (256 samples each of COS and SIN). Thus in a record the status information is repeated 8 times. The time interval between two PERTEC records is 12288 msec.

In order to verify the functioning of the DAS, we need to check this status information at the end of each day, so that any malfunctioning of the system can be rectified immediately without loss of data.

CODE DESCRIPTION

The program consists of the following routines - COPYMD (main program) ICONS, SETIND, RMOVE, HEADER, SYNC, DATED.C, BEAMDC, SOURCE, DKFILE and PRI (See the flow chart in Fig. 4). These routines have the following functions:

(1) COPYMD is the main driver routine. This routine initializes the program parameters such as open files, open magnetic tape unit, initializes the array and the decoding procedure.

(2) ICONS finds the sync-word constant in the beginning of the program. This is then used for comparison with what is decoded from the magnetic tape.

(3) SETIND initializes the magnetic tape operations for industry compatible mode. The IPS data is recorded in a byte mode on 9-track magnetic tape. By this routine we set DEC-10 to work in a 32-bit word machine.

(4) RMOVE moves the given number of bits from a given starting point of a DEC word, to the other DEC word, from the beginning of the specified position of this new word. This is used to assemble the 12 bit data word of COS and SIN and to form the Status-digits. This has been written in MARCO to have highest efficiency.

(5) HEADER decodes the status information of IST, day, station name, beam number, source name etc. using:

(a) SYNC - which checks the Sync. word first.

(b) DATEDC - which decodes the date and IST upto millisecond.

(c) BEAMDC - which decodes the beam number from the beam code.

(d) SOURCE - which decodes the source name, Station name, SDT on time from the status digits.

(6) DKFILE - once data is decoded, it is stored on a Diskfile. This is done in a recordwise manner. A label on top of the disk record is put to show the observing station, source name, beam number etc.

(7) PRI - This prints out the first line of each record of the disk file. So that the DAS operation could be checked.

A listing of the code together with detailed comments and a sample output is shown at the end. Using part of this program one can also print the values in HEX-DECIMAL FORMAT which is used to check the operation of PERTEC-magnetic tape recorder.

Acknowledgements

We would like to express our gratitude to Prof. R.V. Bhonsle and Dr. S.K. Alurkar for their encouragement. We would like to thank Mr. G.G. Dholakia for useful discussions during the development of the program and for timely help at various stages. Thanks are also due to Dr. S.S.Degaonkar for his valuable suggestions in preparing this note.

References

1. Alurkar S.K., Bhonsle R.V., Sharma R., Bobra A.D., Sohanlal, Nirman N.S., Venat P. and Sethia G.: "A Correlation interferometer for Interplanetary scintillation observations", JIETE, 28, 577, 1982.

2. TOPS-10/20 FORTRAN Language manual AA-N383A-TK Digital Equipment Corporation.

3. DEC System-10 Processor Reference Manual AA-H391A-TK Digital Equipment Corporation.

4. TOPS-10 Monitor Calls Volume 1 AA-0974D-TB Digital Equipment Corporation.

5. TOPS-10 Monitor Calls Volume 2 AA-K039A-TB Digital Equipment Corporation.

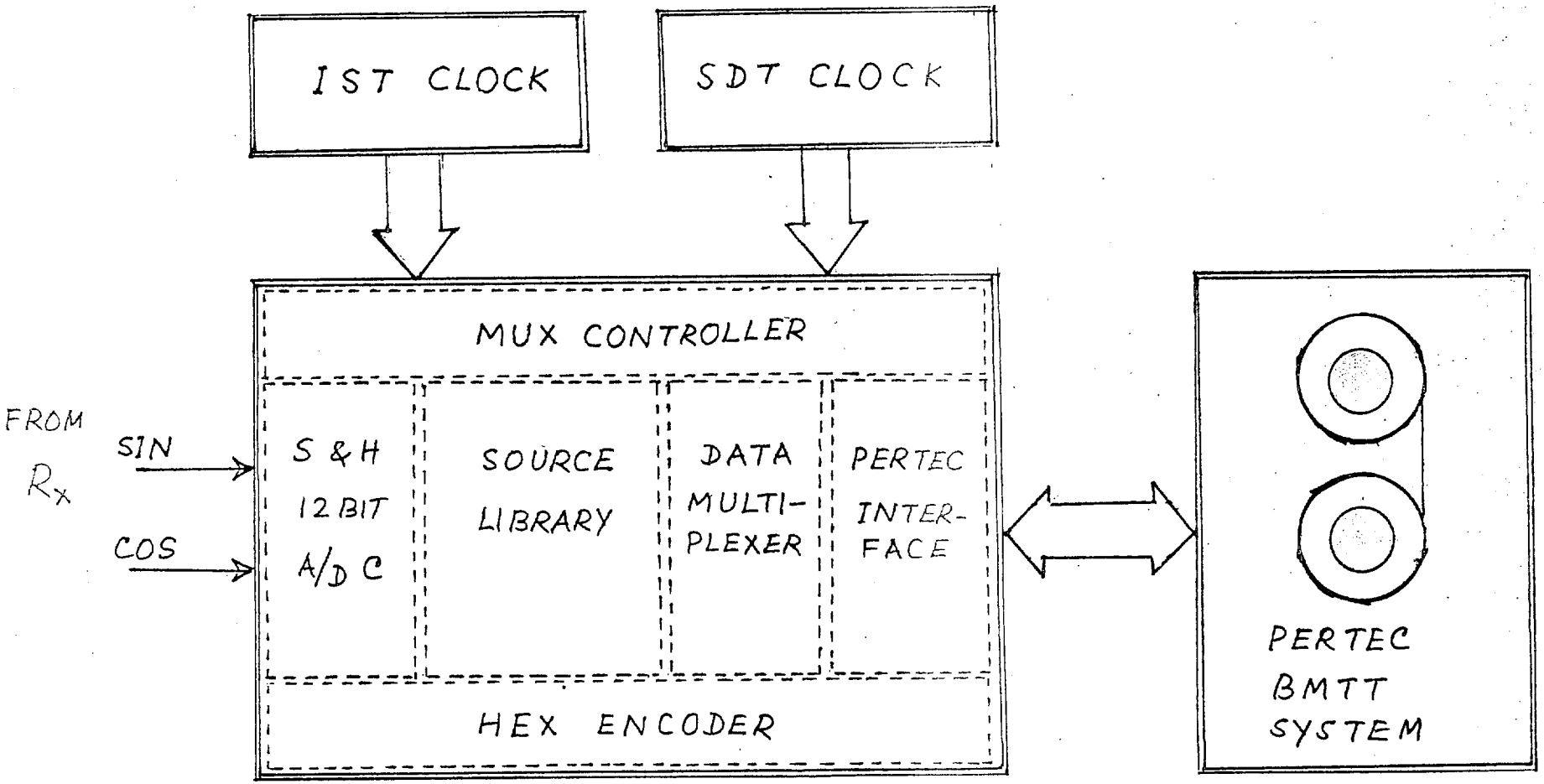


Fig.1- IPS DAS BLOCK DIAGRAM

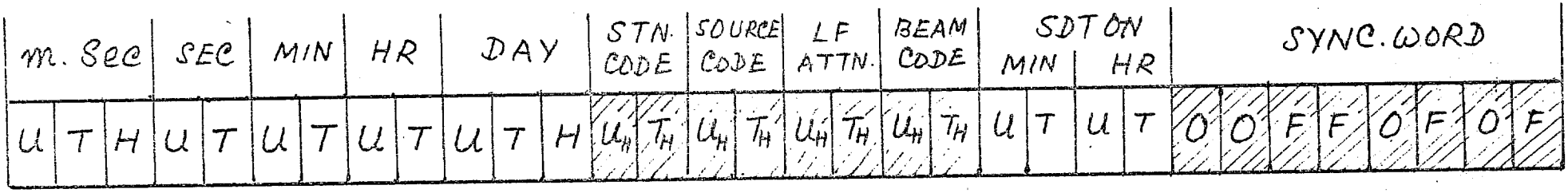


Fig 2-IPS SYSTEM STATUS INFORMATION
(DAS FORMAT)

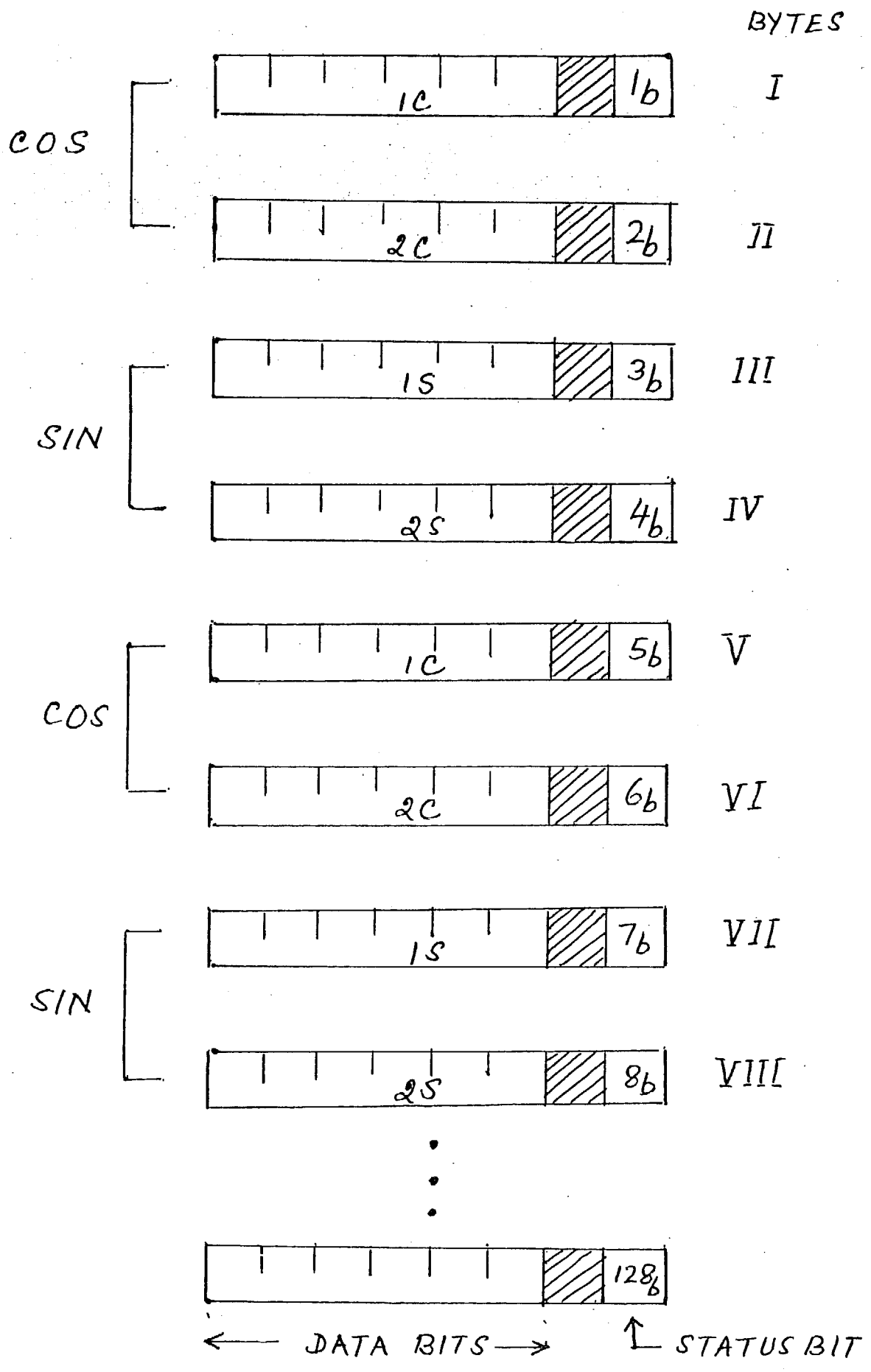


Fig 3 - IPS DATA AND STATUS INFORMATION (PERTEC FORMAT)

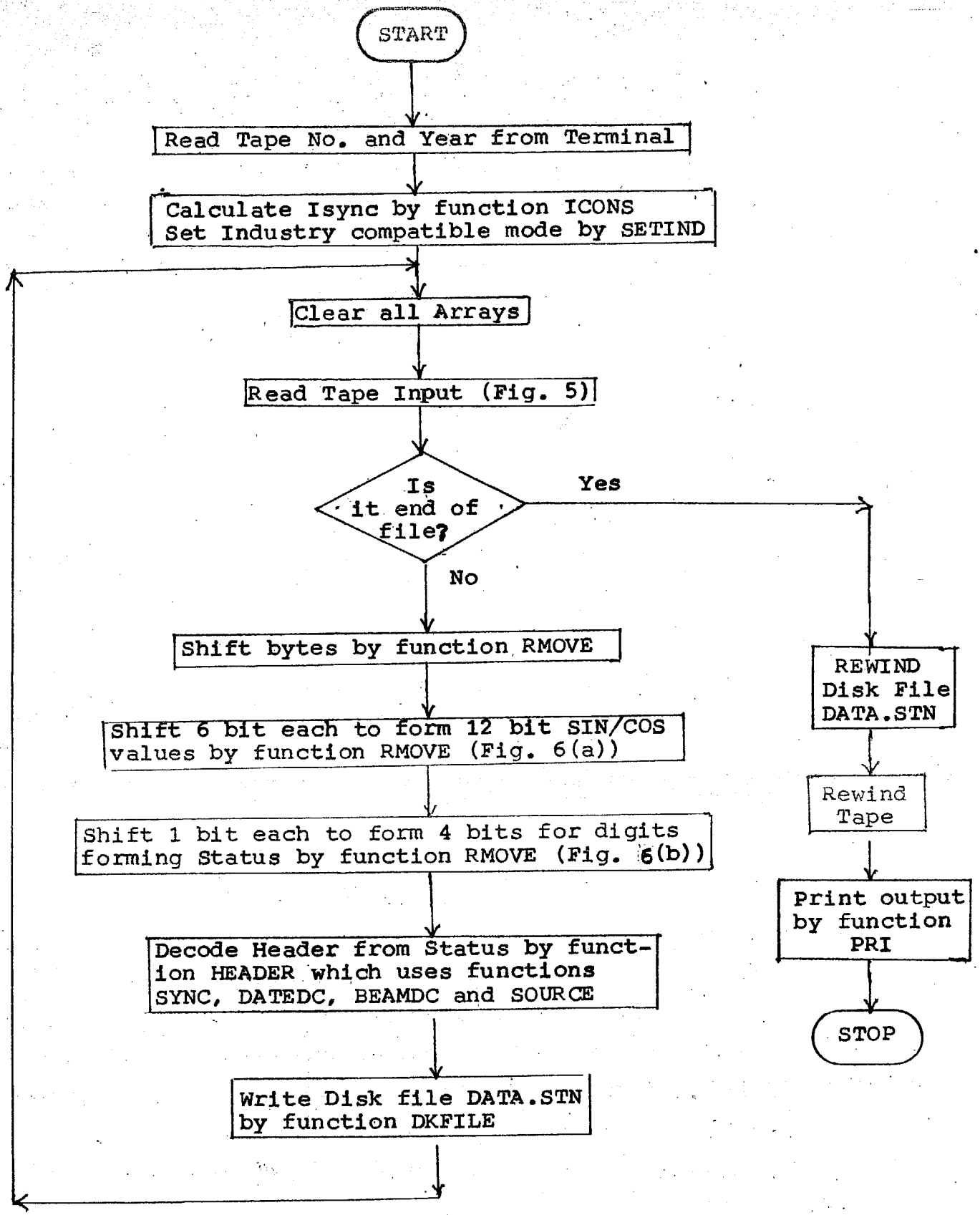


Fig. 4. Schematic Flow Chart of the IPS Data Decoding Program.

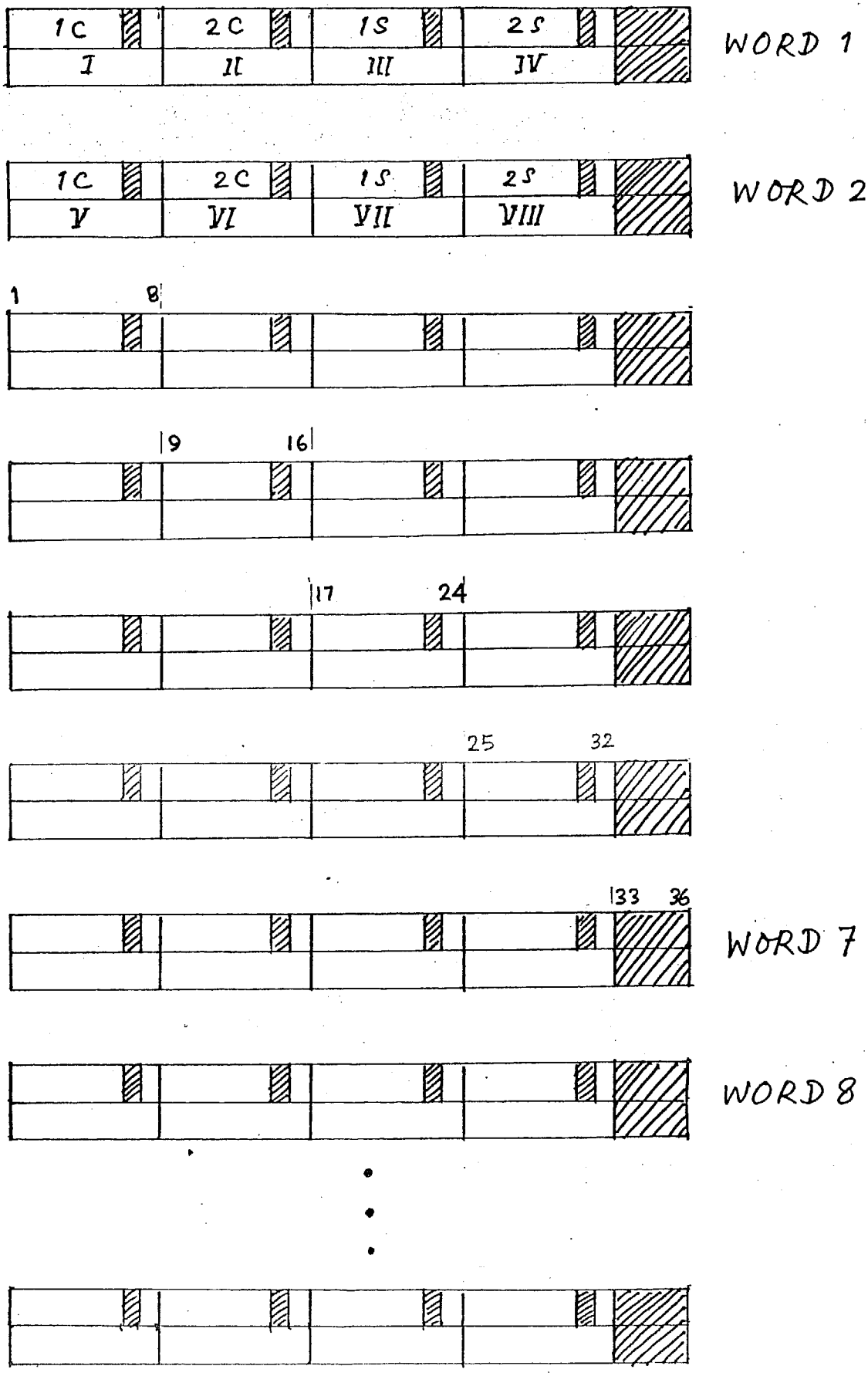
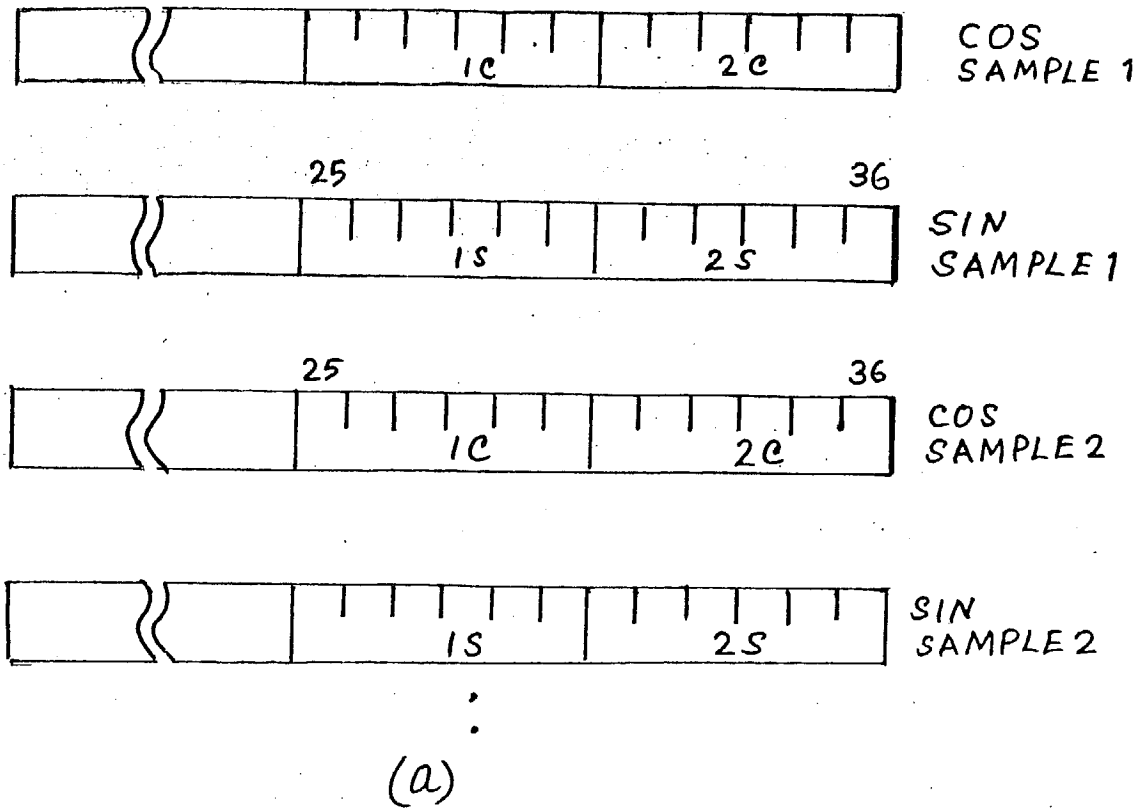
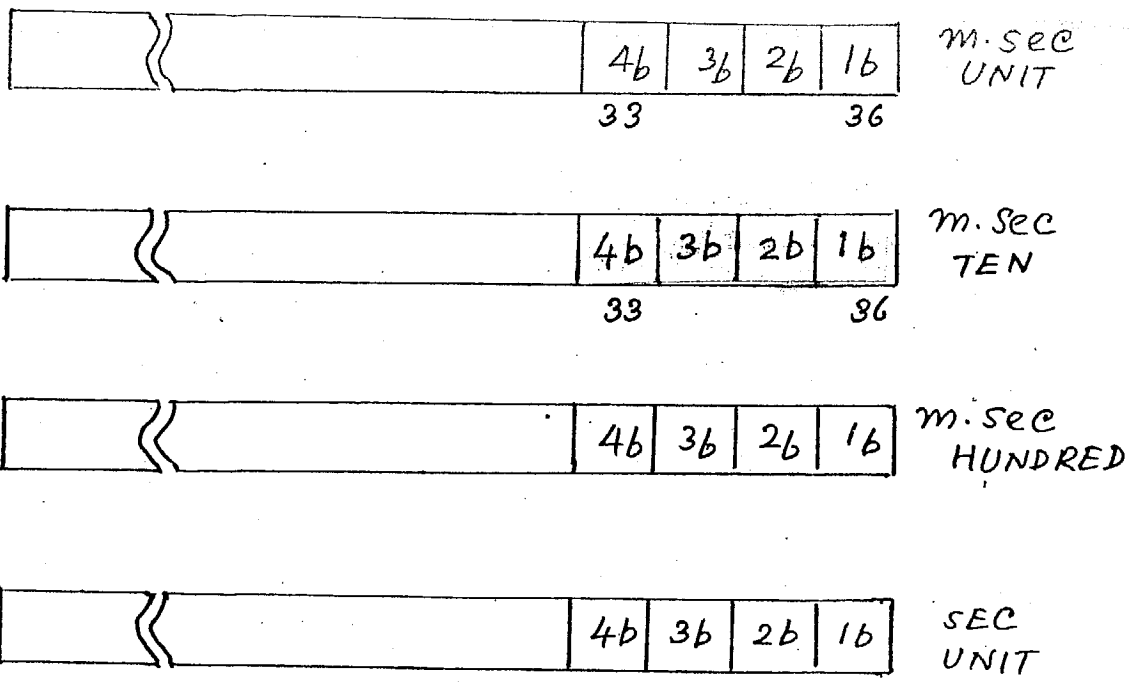


Fig. 5 DATA INPUT (DEC 10)



(a)



(b)

Fig. 6 DATA DECODING (DEC 10)

Observations.....RAJKOT

Source## 3C147 Beam## 13R Lfa## 0 sdt on-time## 5h 11m

12- 6-1986	10h 53m 13s 105msec	1974	2102	1979	2107	1979	2107	1974
12- 6-1986	12h 25m 12s 337msec	1967	2095	1969	2097	1973	2101	1975
12- 6-1986	12h 25m 24s 625msec	1863	2119	1863	2119	1867	2123	1867
12- 6-1986	12h 25m 36s 913msec	1965	2093	1975	2103	1971	2099	1975
12- 6-1986	12h 25m 49s 201msec	1935	2127	1937	2129	1935	2127	1937
12- 6-1986	12h 26m 1s 489msec	1901	2093	1903	2095	1903	2095	1899
12- 6-1986	12h 26m 13s 777msec	1957	2149	1953	2145	1959	2151	1955
12- 6-1986	12h 26m 26s 65msec	1861	2117	1919	2111	1858	2114	1919
12- 6-1986	12h 26m 38s 353msec	1947	2139	1879	2135	1881	2137	1947
12- 6-1986	12h 26m 50s 641msec	1885	2141	1871	2127	1873	2129	1870
12- 6-1986	12h 27m 2s 929msec	1874	2130	1877	2133	1867	2123	1865
12- 6-1986	12h 27m 15s 217msec	1899	2155	1891	2147	1886	2142	1885
12- 6-1986	12h 27m 27s 505msec	1909	2165	1909	2165	1906	2162	1903
12- 6-1986	12h 27m 39s 793msec	1907	2163	1903	2159	1899	2155	1906
12- 6-1986	12h 27m 52s 81msec	1963	2155	1907	2163	1907	2163	1918
12- 6-1986	12h 28m 4s 369msec	1857	2177	1861	2181	1859	2179	1863
12- 6-1986	12h 28m 16s 657msec	1865	2185	1867	2187	1873	2193	1879
12- 6-1986	12h 28m 28s 945msec	1859	2179	1915	2171	1909	2165	1911
12- 6-1986	12h 28m 41s 233msec	1863	2183	1919	2175	1858	2178	1865
12- 6-1986	12h 28m 53s 521msec	1859	2179	1863	2183	1919	2175	1863
12- 6-1986	12h 29m 5s 809msec	1857	2177	1858	2178	1861	2181	1867
12- 6-1986	12h 29m 18s 97msec	1879	2199	1879	2199	1883	2203	1882
12- 6-1986	12h 29m 30s 385msec	1873	2193	1869	2189	1867	2187	1877
12- 6-1986	12h 29m 42s 673msec	1875	2195	1875	2195	1867	2187	1863
12- 6-1986	12h 29m 54s 961msec	1903	2223	1903	2223	1913	2233	1914
12- 6-1986	12h 30m 7s 249msec	1901	2221	1893	2213	1893	2213	1893
12- 6-1986	12h 30m 19s 537msec	1906	2226	1910	2230	1903	2223	1901
12- 6-1986	12h 30m 31s 825msec	1902	2222	1905	2225	1897	2217	1898
12- 6-1986	12h 30m 44s 113msec	1911	2231	1915	2235	1914	2234	1919
12- 6-1986	12h 30m 56s 401msec	1859	2243	1915	2235	1911	2231	1917
12- 6-1986	12h 31m 8s 689msec	1810	2258	1873	2257	1814	2262	1807
12- 6-1986	12h 31m 20s 977msec	1917	2237	1911	2231	1909	2229	1911
12- 6-1986	12h 31m 33s 265msec	1851	2235	1795	2243	1794	2242	1919
12- 6-1986	12h 31m 45s 553msec	1809	2257	1873	2257	1877	2261	1877
12- 6-1986	12h 31m 57s 841msec	1799	2247	1803	2251	1799	2247	1803
12- 6-1986	12h 32m 10s 129msec	1907	2291	1907	2291	1901	2285	1903

DECODED DATA

.....OPERATIONAL DETAILS.....

```
*Mount your tape on magnetic tape unit with IPS Logical name.
*Delete DATA.THA,DATA.RAJ _DATA.SUR, if exist.
*RUN COPYMD[342,107,IPS]
* Tape-Name : tname     ;tname is the tape name.
* Year      : n                  ;n is the year of data.
*
*
*After execution it will generate a out=put data file on disk -
*                                  a print=out for each record.
*Rewind and Unload the tape.
*
```

.....PROGRAMME LISTING.....

```
*
*
*
c       This is A FORTRAN  program to decode and copy
c       the ips data from magnetic tape to disk.
c       FILE NAME::::     COPYMD.FOR
c       BY:::::::::::     A.D. BOBRA
c       Data first read from the tape and decoded
c       to give you the the status and the
c       integer values of the data points.
c       The disk files are created depending on the
c       station code decoded from the data.
c       DATA.THA          DATA.RAJ          DATA.SUR
c       subroutines required
c       ICONS,RMOVE,HEADER,SYNC,DATEDC,BEAMDC,SOURCE,
c       DKFILE,PRI.
c
        dimension ia(2048),ib(512),ic(2048),id(512)
        dimension ie(512),istat(10)
        character*3 bm
        character*9 src,obs,tape
        common/head/idy,imon,iyer,ih,im,isec,imsc,src,
     1  obs,bm,lfa,isdh,isdm
        common/flag/nflg(16)
        nrec=1024
10      type15  ;accept20,tape  ;type25 ;accept*,iyer
15      format(' Tape-Name : ',$)
20      format(a)
25      format(' Year      : ',$)
c
c
c
c       now generat the sync word
c
        call icons(isync)
c
```

```
c       now position the tape and read the data
c
        n=0
        jj=nrec/4
        open(unit=17,device='IP5',mode='dump')
        call setind
50      n=n+1
        do i=1,2048
        ia(i)=0 ;ic(i)=0
        end do
        do i=1,512
        ib(i)=0 ;id(i)=0          ;ie(i)=0
        end do
        read(unit=17,err=999,iostat=istat(1),end=888)ia
        if(istat(1).eq.1)then
        goto 888
        end if
c
c       now decode the data using RMOVE and store the
c       "header data" in the   ib array
c       then assemble the 'cos' and 'sin' values in
c       the  'id' and  'ie'  array respectively.
c
        do k=1,jj
        kk=(k-1)*4
        call Rmove(ia(k),1,8,ic(kk+1),29)
        call Rmove(ia(k),9,8,ic(kk+2),29)
        call Rmove(ia(k),17,8,ic(kk+3),29)
        call Rmove(ia(k),25,8,ic(kk+4),29)
        call rmove(ia(k),1,6,id(k),25)
        call rmove(ia(k),9,6,id(k),31)
        call rmove(ia(k),17,6,ie(k),25)
        call rmove(ia(k),25,6,ie(k),31)
        end do
c
c       now data is available in the byte mode
c       transfer in the  'ib' - array  for digits.
c
        l=1
        do j=1,JJ
        k=(j-1)*4
        call Rmove(ic(K+1),36,1,ib(l),36)
        call Rmove(ic(K+2),36,1,ib(l),35)
        call Rmove(ic(K+3),36,1,ib(l),34)
        call Rmove(ic(K+4),36,1,ib(l),33)
        l=l+1
        end do
c       now      ib       header
c                id       cos
c                ie       sin
        call header(ib,isync)
        call dkfile(ib,id,ie,n,jj,tape)
c
c       if eof not detected then return to top
c
```

```
      goto 50
888   write(5,200)istat(1)
200   format(5x,'eof  detected..',i3)
      goto 500
999   write(5,300)istat(1)
300   format(5x,'read error detected..',o9)
500   continue
      rewind (55)
      call pri
      close (55)
      stop
      end
C     *****   *****   *****   *****
      subroutine icons(isync)
c
c     This is to find the constant
c     equal to [00FF0F0F] in  hex.
c
      isync=2**23+2**22+2**21+2**20+2**19+2**18+
     1      2**17+2**16+2**11+2**10+2**9+2**8+
     1      2**3+2**2+2**1+2**0
      return
      stop
      end
c     *****   *****   *****   *****
      subroutine header(ic,isync)
c     This is the subroutine to decode the
c     header of the  ips  data
c     giving the source,day,date,beam,station
c     and the  IST  time.
c     12 bit  - 3 digit        1-3   milli-second
c      8      - 2              4-5   second
c      8      - 2              6-7   minute
c      8      - 2              8-9   hour
c     12      - 3              10-12 day
c      8      - 2              13-14 station code
c      8      - 2              15-16 source code
c      8      - 2              17-18 low freq. attenuation
c      8      - 2              19-20 beam code
c      8      - 2              21-22 SDT on time minute
c      8      - 2              23-24              hour
c     32      - 8              25-32 Sync  word [00FF0F0F]
c
      dimension ic(512)
      character*9 obs,src
      character*3 bm
      common/head/idy,imon,iyer,ih,im,isec,imsc,
     1 src,obs,bm,lfa,isdh,isdm
      common/flag/nflg(16)
c     First check the Sync.Word.
      icyc=0
10    icyc=icyc+1
      call sync(ic,icyc,isync)
      if(icyc.gt.8)goto 60
      if(nflg(1).eq.0)then
```

```
      goto 20
      else
      goto 10
      end if
c     Now decode the date and time.
20    continue
      call datedc(ic,icyc)
      if(nflg(2).eq.1)then
      goto 10
      else
      goto 30
      end if
c     Now decode the beam number.
30    continue
      call beamdc(ic,bm,icyc)
      if(nflg(3).eq.1)then
      goto 10
      else
      goto 40
      end if
c     Now decode the Source.
40    continue
      call source(ic,icyc)
      if((nflg(4).eq.1).or.(nflg(5).eq.1).or.(nflg(6).eq.1))then
      goto 10
      else
      goto 60
      end if
60    continue
      return
      stop
      end
C     *****   *****   *****   *****
      subroutine sync(ic,icyc,isync)
c
c     This is to decode the synchronization word
c     the word is  00FF0F0F
C     If the sync.word is wrong iflg(1) is set to
c     '1' and returned to main program else it is '0'.
c
      dimension ic(512)
      common/flag/iflg(16)
      iflg(1)=0
      ni=(icyc-1)*32
      j1=ni+32
      j2=ni+31
      j3=ni+30
      j4=ni+29
      j5=ni+28
      j6=ni+27
      j7=ni+26
      j8=ni+25
c
c     convert the sync word value
c
```

```
      k1=ic(j1)
      k2=ic(j2)*(16**1)
      k3=ic(j3)*(16**2)
      k4=ic(j4)*(16**3)
      k5=ic(j5)*(16**4)
      k6=ic(j6)*(16**5)
      k7=ic(j7)*(16**6)
      k8=ic(j8)*(16**7)
c
c     now sum then up
c
      jj=k1+k2+k3+k4+k5+k6+k7+k8

      if(jj.eq.isync)then
      return
      else
      write(5,1000)jj
1000  format(5x,'sync word wrong',4x,8z1,1)
      iflg(1)=1
      end if
      return
      stop
      end
c     *****   *****   *****   *****
      subroutine datedc(ic,ncyc)

c
c     This is the subroutine to decode the date and
c     IST time from the header block
c
c     information is available in the following
c     data words as one digit information.
c
c
c
c                     day     12      hundred
c                             11      tenth
c                             10      unit
c                     hour     9      tens
c                              8      units
c                     min      7      tens
c                              6      units
c                     sec      5      tens
c                              4      units
c                     m-sec    3      hundreds
c                              2      tens
c                              1      units
c
c     if in the first cycle the date etc not all right
c     one should look into the next block
c
c             (ncyc-1)*32+the value
c
c
c     If any of the checks are not within the limit
c     specified then set the iflg(2)='1' else '0'.
c
```

```
      dimension ic(512),l(12),month(13)
      character*3 beam
      character*9 source,obs
      common/flag/nflg(16)
      common/head/idy,imon,iyer,ih,im,isc,imsc,source,
     1 obs,beam,lfa,isdh,isdm
      data month/00,31,59,90,120,151,181,212,243,
     1 273,304,334,365/
      nflg(2)=0
5     ni=(ncyc-1)*32
      do k=1,12
      l(k)=ic(ni+k)
      end do
c
      iday=l(10)+l(11)*10+l(12)*100
      ih=l(8)+l(9)*10
      im=l(6)+l(7)*10
      isc=l(4)+l(5)*10
      imsc=l(1)+l(2)*10+l(3)*100
c
      i=1
10    continue
      if(iday.gt.month(i))then
      i=i+1
      goto 10
      else
      goto 110
      end if
110   continue
      ni=i-1
      imon=ni
      idy=iday-month(ni)
      if((idy.gt.31).or.(ih.gt.24).or.(im.gt.60).or
     1 .(isc.gt.60))then
      nflg(2)=1
      return
      else
      goto 120
      end if
120   continue
      return
      stop
      end
C     *****   *****   *****   *****
      subroutine beamdc(ic,bm,ncyc)
c     This is the program to decode the beam
c     from the header data.
c
c     data in  19  unit
c              20  tens
c     If in first cycle in not all right then
c     process the 2nd  cycle the data
c     in then available in the
c
c                     ( ncyc-1)*32+19 unit
```

```
                    ( ncyc-1)*32+20  tens

  If the ckecks are not within the specified
  limits then set iflg(3)='1', else '0'.

  dimension code(32),ic(512)
  character*3 bm,code
  common/flag/nflg(16)
  data code/'16L','15L','14L','13L','12L','11L','10L',
 1 '9L','8L','7L','6L','5L','4L','3L','2L',
 1 '1L','1R','2R','3R','4R','5R','6R',
 1 '7R','8R','9R','10R','11R','12R',
 1 '13R','14R','15R','16R'/

  nflg(3)=0
  il=(ncyc-1)*32+19
  ih=(ncyc-1)*32+20
  i=ic(il)+ic(ih)*16
  ni=i+1
  if(ni.gt.32)then
  write(5,100)ni,ncyc
  format(5x,'beam-code wrong/cyc = ',2i4)
  nflg(3)=1
  return
  end if

  bm(1:3)=code(ni)
  return
  stop
  end
  *****    *****    *****    *****
  subroutine source(ic,ncyc)


  This is the sub-routine to decode the source
  name,station code,low-freq attenuation, and
  SDT  on time in hours and minutes


  source code in          15      unit
                          16      tens  in hex

  lf attenuation          17      unit
                          18      tens     decimal

  station                 13      unit
                          14      tens      hex

  sdt on time             21      minute   unit
                          22               tens
                          23      hour     unit
                          24               tens


  station          Thaltej(Old-unit)        c1       -193
                   Thaltej(New-unit)        d2       -210
                   Surat                    cd       -205
                   Rajkot                   c9       -201
```

```
c
      dimension ic(512),stn(4),sourc(256),istncd(4),ix(32)
      character*3 bm
      character*9 sourc,src,stn,obs
      common/flag/iflg(16)
      common/head/id,imn,iyer,ih,im,isc,imsc,src,obs,
     1 bm,lfa,isdh,isdm
      data stn/'Thaltej','Thaltej','SURAT','RAJKOT'/
      data sourc/'3C2','3C9','3C13','3C15','3C19','3C20',
     1 '3C22','3C23','3C26','3C33','3C34','3C36',
     1 '3C39','3C42','3C43','3C44',
     1 '3C47','3C48','3C49','3C53','3C54','3C55',
     1 '3C56','3C65','3C67','3C68.1','3C68.2','3C71','3C74',
     1 '3C82','3C84','3C89','3C90','3C91','3C93.1','3C96',
     1 '3C98','3C99','3C103','3C105','3C108','3C111','3C115',
     1 '3C119','3C120','3C123','3C124','3C125','3C128',
     1 '3C133','3C134','3C136','3C137','3C138','3C144',
     1 '3C147','3C152','3C153','3C154','3C158','3C159',
     1 '3C161','3C169.1','3C172','3C173','3C175','3C175.1',
     1 '3C181','3C183','3C186','3C190','3C191','3C194',
     1 '3C194.1','3C196','3C196.1','3C197','3C197.1','3C198',
     1 '3C202','3C208','3C210','3C211','3C212','3C213.1',
     1 '3C216','3C217','3C218','3C219','3C220.2','3C222',
     1 '3C223','3C225','3C226','3C228','3C230','3C234',
     1 '3C235','3C236','3C237','3C238','3C239','3C241',
     1 '3C245','3C247','3C251','3C252','3C254','255',
     1 '3C256','3C263.1','3C264','3C265','3C266','3C267',
     1 '3C268','3C268.8','3C270','3C270.1','3C272','3C273',
     1 '3C274','3C275','3C275.1','3C277.3','3C279','3C280',
     1 '3C280.1','3C284','3C286','3C287','3C288.1','3C289',
     1 '3C291','3C293','3C293.1','3C294','3C295','3C297',
     1 '3C298','3C299','3C300.1','3C303','3C304','3C310',
     1 '3C311','3C315','3C316','3C317','3C318','3C318.1',
     1 '3C321','3C324','3C326.1','3C327','3C331','3C336',
     1 '3C337','3C338','3C342','3C344','3C345','3C346',
     1 '3C348','3C349','3C352','3C353','3C354','3C355',
     1 '3C356','3C359','3C364','3C368','3C372','3C380',
     1 '3C400.1','3C405','3C409','3C410','3C411','3C431',
     1 '3C432','3C433','3C437.1','3C438','3C441','3C442',
     1 '3C445','3C446','3C449','3C452','3C453','3C454',
     1 '3C454.3','3C455','3C456','3C459','3C461','3C463',
     1 '3C465','3C466','3C468','3C470','SS433','CTA21',
     1 'CP0950','CP1133','CP1919','VFP','5252A','GG1d4',
     1 'IRC20431','IRC20540','NGC2175','IRC20127'/
c
c
c
      inc=(ncyc-1)*32
      istncd(1)=193
      istncd(2)=210
      istncd(3)=205
      istncd(4)=201
      do i=13,24
      ix(i)=inc+i
      end do
```

```
c
c
      do i=4,8
      iflg(i)=0
      end do
c
c     now data available in the ic array
c           station code
c
      i1=ix(13)
      i2=ix(14)
      jj=ic(i1)+ic(i2)*16
      i=1
10    continue
      if(jj.eq.istncd(i))then
      obs=stn(i)
      goto 1000
      else if(i.gt.4)then
      iflg(4)=1
      write(5,100)jj,ncyc
100   format(3x,'station code wrong ',2i5)
      goto 1000
      else
      i=i+1
      goto 10
      end if
1000  continue
c
c     source code
c
      i1=ix(15)
      i2=ix(16)
      jj=ic(i1)+ic(i2)*16
      src=sourc(jj)
c
c
c     Now low-freq attenuation
c
      i1=ix(17)
      i2=ix(18)
      jj=ic(i1)+ic(i2)*10
      lfa=jj
      if(jj.gt.16)then
      write(5,200)jj,ncyc
      iflg(5)=1
200   format(3x,'lf attenuation worng',2i5)
      goto 2000
      end if
2000  continue
c
c
c     sdt-on   time  in decimal
c
c
      i1=ix(21)
```

```
      i2=ix(22)
      i3=ix(23)
      i4=ix(24)
      jj=ic(i1)+ic(i2)*10
      kk=ic(i3)+ic(i4)*10
      isdm=jj
      isdn=kk
      if((isdm.gt.60).or.(isdn.gt.24))then
      iflg(5)=1
      write(5,300)jj,kk,ncyc
300   format(3x,'sdt on time wrong  ',3i6)
      goto 3000
      end if
3000  continue
      return
      stop
      end
C     *****   *****   *****   *****
      subroutine dkfile(ic,id,ie,nr,nrec,tape)
c             this file is used to create the
c             disk file after every thing is ok.
c
c             data is available in the common head
c                  and the id,  ie,   array
c             nr...  represents the record no.
c                  this is necessary so that the
c                  common header information can be
c                  kept in the begining and need
c                  not be repeated every time.
c
c             tape..  this the tape name or number
c
c             nrec..  this is the record length in the
c                  origional data stream.
      dimension ic(512),id(512),ie(512)
      character*3 bm
      character*9 src,obs,tape,nfile
      common/head/iday,imon,iyer,ih,im,is,imsc,src,obs,bm,
     1 lfa,isdn,isdm
c
c
c     now first if this is the first record then open
c             the new file,and write the standard information
c
      ir=nr
      if(ir.eq.1)then
      write(nfile,100)obs
100   format('data.'a3)
      TYPE *,'NFILE.....',NFILE
      open(unit=55,device='dskc',file=nfile,status='new',
     1 form='formatted')
      write(55,200)tape,obs
200   format(20x,'Tape label.....',a9,/,20x,
     1 'Observations.....',a9,//)
      write(55,300)src,bm,lfa,isdn,isdm
```

```
300     format(3x,'Source     ',a9,3x,'Beam     ',a3,3x,'Lfa    ',
     1  i3,3x,'sdt  on-time     ',i3,'h',1x,i3,'m',//)

        end if
c
c
c       now continuous data stream to be written
c
c
        write(55,400)iday,imon,iyer,ih,im,is,imsc,(id(i),
     1  ie(i),i=1,nrec)
400     format(3x,i2,'-',i2,'-',i4,3x,i2,'h ',i2,'m ',
     1  i2,'s ',i3,'msec ',5x,512(2i5))
        return
        stop
        end
C       *****   *****   *****   *****
        SUBROUTINE PRI
C       This is the program to take the print of the
C       This is the program to take the print of the
c       data for the reference for the ips files
c       When executed the program asks for the file
c       after supplying the correct file the data
c       is printed on the lpt.
c       THIS PROGRAM IS WRITTEN BY M.S. PATEL OF COMPUTER CENTER.
c
c
        character*82 aa
10      read(55,20,end=100)aa   ;print25,aa     ;goto10
20      format(a)
25      format(1x,a)
100     stop    ;end
C
C       *****   *****   *****   *****
C
;THIS PROG. MOVES SPECIFIED NO. OF BIT(S) FROM THE INPUT WORD
;TO SPECIFIED OUTPUT WORD
;       ************************************************************
;        *        BY   MRS B H SHAH     (COMP. CENTER)              *
;       *                                                           *
;       * CALL RMOVE(X,SX,NX,Y,SY)                                  *
;       * ALL ARE INTEGER VALUE                                     *
;       * X      INPUT WORD X                                       *
;       * SX     STARTING BIT POSITION IN INPUT WORD X              *
;       * NX     NO. OF BITS  TO BE MOVED FROM INPUT WORD X         *
;       * Y      GIVEN OUTPUT WORD                                  *
;       * SY     STARTING BIT POSITION IN Y                         *
;       *                                                           *
;       ************************************************************
RMOVE::         X=1     ;INPUT WORD X
        SX=2    ;STARTING BIT POSITION IN INPUT WORD X
        NX=3    ;NO. OF BITS  TO BE MOVED FROM INPUT WORD X
        NX1=12
        Y=4     ;GIVEN OUTPUT WORD
        SY=5    ;STARTING BIT POSITION IN Y
```

```
        SY1=11
        RMB=10
SAVREG:         MOVEM   17,SAVE+17      ;TO SAVE REG.
        MOVEI   17,SAVE
        BLT     17,SAVE+16
        MOVE    X,@(16)
        MOVE    SX,@1(16)
        MOVE    NX,@2(16)
        MOVE    NX1,NX
        MOVE    Y,@3(16)
        MOVE    SY,@4(16)
        MOVE    SY1,SY
;       FOLLOWING 4 INSTRUCTIONS CHECKS THE GIVEN VALUE
;       [SX + NX -1 ] < 37
        MOVE    7,SX
        ADD     7,NX
        SUBI    7,1
        CAIL    7,D37
        JRST    ERR
        SUBI    7,D36
        MOVE    RMB,7   ;REMAINING BITS TO THE RIGHT
;       CHECKING FOR THE GIVEN VALUE SY -NX
;       [ SY + NX -1 ] MUST BE < 37
        MOVE    7,SY
        ADD     7,NX
        SUBI    7,1
        MOVEM   7,TEMP
        CAIL    7,D37
        JRST    ERR
        SUB     7,7
        MOVE    6,X
        LSH     6,(RMB) ;REMOVE REMAINING BITS (RIGHT)
        MOVN    3,3
        ROTC    6,(3)
        SUBI    5,1
        MOVN    5,5
        LSH     7,(5)
DDB1:   MOVEM   7,ANS1
        MOVE    7,SY1
        SUBI    7,1
DDB2:   MOVEM   7,SHF1
        MOVEI   7,D36
        SUB     7,NX1
        MOVN    7,7
DDB3:   MOVEM   7,SHF2
        MOVEI   7,D36
        SUB     7,TEMP
DDB4:   MOVEM   7,SHF3
        SETO    7,
        MOVE    1,SHF1
DDB5:   LSH     7,(1)
        MOVE    1,SHF2
        LSH     7,(1)
        MOVE    1,SHF3
        LSH     7,(1)
```

```
        SETO    6,
        XOR     6,7
        AND     6,Y
        MOVE    1,ANS1
        OR      6,1
        MOVEM   6,@3(16)
RESTOR:         MOVSI   17,SAVE
        BLT     17,17
        POPJ    17,
RET:    POPJ    17,
ERR:    OUTSTR  [ASCIZ 'ERROR IN INPUT -JOB CANCELED ']
;       EXIT
        JRST    RESTOR
SAVE:   BLOCK   20
ANS1:   BLOCK   1
SHF1:   BLOCK   1
SHF2:   BLOCK   1
SHF3:   BLOCK   1
TEMP:   BLOCK   1
        END
C
C       ************************************.
C
C
ENTRY   SETIND
        MOVE
SETIND:         MOVE 0,[3,,BLK]
        TAPOP.
        HALT
        POPJ 17,
BLK:    2007
        SIXBIT/IPS/
        2
        END
C       ************************************************************
C                           ************************
C                                   ********
```