

1475	1443	0067	-200-1	
1476	1444	0346	WBR SEPCLK T	
1477	1445	1450	.+3	
1478	1446	0216	JUMP F3	/TIMING ERROR
1479	1447	1667	TIMERR	
1480				
1481	1450	0156	BR DEQSR7 T	/WHAT IS SEP DATA?
1482	1451	1755	NOTYET	/ZERO, GO DETERMINE IF TO MANY STARTS
1483				
1484	1452	2164	BR MCEQSR F	/ONE, MISSING CLOCK?
1485	1453	1673	BADSRT	/YES, SHOULDN'T HAVE BEEN
1486				
1487	1454	0057	PRECRC	/JAM 1ST TWO BITS OF CRC
1488	1455	0056	CRC ONE	
1489	1456	2056	CRC ONE	
1490				
1491	1457	0070	LCT	/WAIT 40 MICRO SECONDS FOR SECOND CELL
1492	1460	0067	-200-1	
1493	1461	0346	WBR SEPCLK T	
1494	1462	1465	.+3	
1495	1463	0216	JUMP F3	
1496	1464	1667	TIMERR	
1497				
1498	1465	0156	BR DEQSR7 T	/DATA SHOULD BE 1, MISSING CLK SHOULD BE T
1499	1466	1673	BADSRT	
1500	1467	0166	BR MCEQSR T	
1501	1470	1673	BADSRT	
1502				
1503	1471	0056	CRC ONE	/JAM 3 MORE CRC BITS
1504	1472	0056	CRC ONE	
1505	1473	0056	CRC ONE	
1506				
1507	1474	0070	LCT	/WAIT FOR THIRD BIT CELL
1508	1475	0067	-200-1	
1509	1476	0346	WBR SEPCLK T	
1510	1477	1502	.+3	
1511	1500	0216	JUMP F3	
1512	1501	1667	TIMERR	
1513				
1514	1502	0154	BR DEQSR7 F	/DATA SHOULD BE 0, MISSING CLK SHOULD BE F
1515	1503	1673	BADSRT	
1516	1504	0164	BR MCEQSR F	
1517	1505	1673	BADSRT	
1518				
1519	1506	2070	LCT	/CLEAR SR
1520	1507	0000	0	
1521	1510	0075	LSR	
1522				
1523	1511	0070	LCT	/WAIT FOR 4TH BIT CELL
1524	1512	0067	-200-1	
1525	1513	0346	WBR SEPCLK T	
1526	1514	1517	.+3	
1527	1515	0216	JUMP F3	
1528	1516	1667	TIMERR	
1529				

1530	1517	0154	BR DEQSR7 F	/DATA SHOULD BE 0, MISSING CLK SHOULD BE F
1531	1520	1673	BADSRT	
1532	1521	0042	LDMD	/NOOP FOR LONG SEP CLOCK
1533	1522	0042	LDMD	/NOOP FOR LONG SEP CLOCK
1534	1523	0164	BR MCEQSR F	
1535	1524	1673	BADSRT	
1536				
1537	1525	0070	LCT	/WAIT FOR FIFTH BIT CELL
1538	1526	0067	-200-1	
1539	1527	0346	WBR SEPCLK T	
1540	1532	1533	.+3	
1541	1531	0216	JUMP F3	
1542	1532	1667	TIMERR	
1543				
1544	1533	0156	BR DEQSR7 T	/DATA SHOULD BE 1
1545	1534	1673	BADSRT	
1546				
1547	1535	0176	BR FLAGO T	/IF FLAG SET FINISH LOOKING FOR DATA AM
1548	1536	1675	DAM	
1549				
1550	1537	0164	BR MCEQSR F	/FINISH IDAM, MISSING CLK SHOULD BE F
1551	1540	1673	BADSRT	
1552				
1553	1541	0256	CRC ONE	/JAM 6TH CRC BIT OF IDAM
1554				
1555	1542	0270	LCT	/WAIT FOR SIXTH BIT CELL
1556	1543	0067	-200-1	
1557	1544	0346	WBR SEPCLK T	
1558	1545	1550	.+3	
1559	1546	0216	JUMP F3	
1560	1547	1667	TIMERR	
1561				
1562	1552	0156	BR DEQSR7 T	/DATA SHOULD BE 1, MISSING CLK SHOULD BE F
1563	1551	1673	BADSRT	
1564	1552	0164	BR MCEQSR F	
1565	1553	1673	BADSRT	
1566				
1567	1554	0042	LDMD	/NOOP FOR LONG SEP CLOCK
1568				
1569	1555	0056	CRC ONE	/JAM 7TH CRC BIT OF IDAM
1570				
1571	1556	0070	LCT	/WAIT FOR SEVENTH BIT CELL
1572	1557	0067	-200-1	
1573	1560	0346	WBR SEPCLK T	
1574	1561	1564	.+3	
1575	1562	0216	JUMP F3	
1576	1563	1667	TIMERR	
1577				
1578	1564	0156	BR DEQSR7 T	/DATA SHOULD BE 1, MISSING CLK SHOULD BE T
1579	1565	1673	BADSRT	
1580	1566	0166	BR MCEQSR T	
1581	1567	1673	BADSRT	
1582				
1583	1570	0054	CRC ZERO	/IDAM FOUND, JAM LAST CRC BIT
1584				

```

1585
1586
1587
1588
1589      /THIS ROUTINE COMPARES THE HEADER TRACK ADDRESS TO THE
1590      /DESIRED TRACK ADDRESS ON THE FLY. IT IS ENTERED AFTER
1591      /FINDING THE IDAM, ERREG BIT 7 IS SET IF AN ERROR IS DETECTED.
1592
1593
1594      1571  2220  HDRCOM, OPEN TARTRK      /TARGET TRACK ADDRESS TO SR
1595      1572  0071  ESP
1596      1573  0075  LSR
1597
1598      1574  0070  LCT                      /SET BIT COUNTER TO 8
1599      1575  0367  -8-1
1600
1601      1576  0144  AGAIN3, BR SEPCLK F      /WAIT FOR BIT CELL
1602      1577  1576  .-1
1603
1604      1600  0156  BR DECSR7 T              /SEP DATA EQUAL TO SR??
1605      1601  1605  .+4                      /NO, TRACK COMPARE ERROR
1606
1607      1602  0074  ROTATE ZERO              /YES, GET NEXT TRACK ADDRESS BIT
1608      1603  0216  JUMP F3
1609      1604  1610  .+4
1610
1611      1605  2210  OPEN ERREG              /SET ERREG BIT 7 TO INDICATE TRACK ERROR
1612      1606  0076  ROTATE ONE
1613      1607  0064  LSP
1614
1615      1612  0055  DATCRC                  /UPDATE THE CRC
1616
1617      1611  0073  ICT                      /INCREMENT AND TEST THE BIT COUNTER
1618      1612  0124  BR COFL F
1619      1613  1576  AGAIN3                  /GO DO NEXT BIT
1620
1621      1614  0070  LCT                      /TRACK COMPARED, SET UP BIT COUNTER FOR 8 BYTE
1622      1615  0367  -8-1
1623
1624      1616  0144  AGAIN4, BR SEPCLK F      /WAIT FOR BIT
1625      1617  1616  .-1
1626
1627      1620  0061  FLAG OFF                /CLEAR FLAG FOR NEXT ROUTINE
1628      1621  0061  FLAG OFF                /NOOP FOR LONG SEP CLK
1629      1622  0061  FLAG OFF                /NOOP FOR LONG SEP CLK
1630      1623  0061  FLAG OFF                /NOOP FOR LONG SEP CLK
1631
1632      1624  0055  DATCRC                  /UPDATE CRC
1633
1634      1625  0073  ICT                      /INCREMENT AND TEST BIT COUNT
1635      1626  0124  BR COFL F
1636      1627  1616  AGAIN4                  /GO DO ANOTHER BIT
1637
1638
1639

```

```

1640      /THIS ROUTINE COMPARES THE HEADER SECTOR ADDRESS WITH THE
1641      /TARGET SECTOR ADDRESS ON THE FLY. IT IS ENTERED FROM
1642      /THE TRACK COMPARE ROUTINE. A MISMATCH WILL SET THE FLAG.
1643
1644
1645      1630  2224  OPEN TARSEC              /TARGET SECTOR ADDRESS TO SR
1646      1631  0071  ESP
1647      1632  0075  LSR
1648
1649      1633  0070  LCT                      /SET UP BIT COUNTER FOR 8 BITS
1650      1634  0367  -8-1
1651
1652      1635  0144  AGAIN5, BR SEPCLK F      /WAIT FOR A BIT
1653      1636  1635  .-1
1654
1655      1637  0156  BR DECSR7 T              /HOW DO THEY COMPARE?
1656      1640  1643  .+3                      /BAD, GO SET THE FLAG
1657
1658      1641  2216  JUMP F3                  /GOOD, SKIP THE ERROR FLAG.
1659      1642  1644  .+2
1660
1661      1643  2262  FLAG ON                  /SET FLAG TO INDICATE MISMATCH
1662
1663      1644  0074  ROTATE ZERO              /BRING UP NEXT BIT
1664
1665      1645  0055  DATCRC                  /UPDATE THE CRC
1666      1646  0073  ICT                      /BUMP THE BIT COUNTER
1667      1647  0124  BR COFL F                /ALL BITS COMPARED?
1668      1650  1635  AGAIN5                  /NO, LOOP BACK
1669
1670      1651  0070  LCT                      /YES, SETUP TO WAIT FOR END OF
1671      1652  0347  -24-1                    /CRC
1672
1673      1653  0144  AGAIN6, BR SEPCLK F      /WAIT FOR BIT
1674      1654  1653  .-1
1675
1676      1655  0074  ROTATE ZERO              /NOOP FOR LONG SEP CLK
1677      1656  0074  ROTATE ZERO              /NOOP FOR LONG SEP CLK
1678      1657  0074  ROTATE ZERO              /NOOP FOR LONG SEP CLK
1679      1660  0074  ROTATE ZERO              /NOOP FOR LONG SEP CLK
1680
1681      1661  2255  DATCRC                  /UPDATE CRC
1682
1683      1662  0073  ICT                      /BUMP THE BIT COUNTER
1684      1663  0124  BR COFL F                /ALL DONE?
1685      1664  1653  AGAIN6                  /NO, LOOP BACK
1686
1687      1665  3226  JUMP F5                  /YES, GO CHECK IF CRC IS ALL ZEROES
1688      1666  2515  CMHCR
1689
1690
1691      1667  0070  TIMERR, LCT              /40 MICROSEC PASSED AND NO SEP CLOCK WAS SEEN
1692      1670  0110  RTIMERR
1693      1671  0226  JUMP F5
1694      1672  2612  GOERDN

```

1695				
1696				
1697	1673	0226	BADSRT, JUMP F5	/POINTER TO BADSTART ON IDAM OR DATA AM
1698	1674	2555	BDSRT	
1699				
1700				
1701				
1702				
1703				
1704				
1705				
1706				
1707				
1708	1675	0166	DAM, BR MCEQSR T	/MISSING CLK SHOULD BE T
1709	1676	1673	BADSRT	
1710				
1711	1677	0054	CRC ZERO	/JAM 6TH CRC BIT OF DATA AM
1712				
1713	1700	0070	LCT	/WAIT FOR SIXTH BIT CELL
1714	1701	0067	-200-1	
1715	1702	0346	WBR SEPCLK T	
1716	1703	1706	+3	
1717	1704	0216	JUMP F3	
1718	1705	1667	TIMERR	
1719				
1720	1706	0164	BR MCEQSR F	/MISSING CLK SHOULD BE F
1721	1707	1673	BADSRT	
1722	1710	0042	LDHD	/NOOP FOR LONG SEP CLK
1723				
1724	1711	0156	BR DEQSR7 T	/IF DATA=0 THEN LOOK FOR DELETED DATA AM
1725	1712	1727	DEL DAT	
1726				
1727	1713	0056	CRC ONE	/JAM 7TH BIT OF DATA AM
1728				
1729	1714	0070	LCT	/WAIT FOR SEVENTH BIT OF DATA AM
1730	1715	0067	-200-1	
1731	1716	0346	WBR SEPCLK T	
1732	1717	1722	+3	
1733	1720	0216	JUMP F3	
1734	1721	1667	TIMERR	
1735				
1736	1722	0056	CRC ONE	/JAM LAST BIT OF DATA AM
1737				
1738	1723	0154	BR DEQSR7 F	/DATA SHOULD BE 1
1739	1724	1742	END DAM	/FLAG IS SET TO INDICATE NORMAL DATA MARK
1740				
1741	1725	0216	JUMP F3	/LAST DATA BIT WAS BAD
1742	1726	1673	BADSRT	
1743				
1744				
1745	1727	0054	DEL DAT, CRC ZERO	/JAM 7TH CRC BIT OF DEL DATA AM
1746				
1747	1730	0070	LCT	/WAIT FOR 7TH CELL OF DEL DATA AM
1748	1731	0067	-200-1	
1749	1732	0346	WBR SEPCLK T	

1750	1733	1736	+3	
1751	1734	0216	JUMP F3	
1752	1735	1667	TIMERR	
1753				
1754	1736	0061	FLAG OFF	/CLR FLAG TO INDICATE DELETED DATA MARK
1755				
1756	1737	0054	CRC ZERO	/JAM LAST CRC BIT OF DEL DATA AM
1757				
1758	1740	0154	BR DEQSR7 F	/DATA SHOULD BE 0
1759	1741	1673	BADSRT	
1760				
1761	1742	0164	END DAM, BR MCEQSR F	/MISSING CLK SHOULD BE F FOR BOTH DATA AMS
1762	1743	1673	BADSRT	
1763				
1764	1744	0222	JUMP F4	/GO PICK UP DATA FIELD
1765	1745	2246	DATA	
1766				
1767				
1768	1746	0046	NOZERO, INCR BAR	/INCREMENT AND TEST PREAMBLE FAILURE COUNT
1769	1747	0160	BR BAROFL F	
1770	1750	1416	TRYAGN+3	/OK, TRY AGAIN FOR A PREAMBLE
1771				
1772	1751	0070	NOXPRAM, LCT	/TOO MANY BITS WITH NO ZEROS
1773	1752	0120	NOXPRAM	
1774	1753	0226	JUMP F5	
1775	1754	2610	GOERDN	
1776				
1777				
1778	1755	0246	NOTVET, INCR BAR	/INCR AND TEST IDAM OR DATA AM START FAILURE COUNT
1779	1756	0042	LDHD	/NOOP FOR LONG SEP CLK
1780	1757	0160	BR BAROFL F	
1781	1760	1442	GETDAM+1	/OK, TRY AGAIN
1782				
1783	1761	0070	NOXIDAM, LCT	/TOO MANY ZEROS WHILE LOOKING FOR START OF
1784	1762	0130	NOXIDAM	/IDAM OR DATA AM
1785	1763	0226	JUMP F5	
1786	1764	2610	GOERDN	
1787				
1788				
1789	1765	0212	PNTROY, JUMP F2	/POINTERS FROM CHECKRDY SUBROUTINE TO ROSTAT ROUTINE
1790	1766	1243	CLRID	
1791	1767	0212	PYSRDY, JUMP F2	
1792	1770	1243	CLRID	
1793				
1794	1771	0212	PNTROY, JUMP F2	/POINTERS FROM CHECK RDY TO INITIALIZE ROUTINE
1795	1772	1006	OKDONE	
1796	1773	0226	JUMP F5	
1797	1774	2631	INTROY	
1798				
1799	1775	0000	0	/OPEN
1800	1776	0000	0	/OPEN
1801	1777	0000	0	/OPEN
1802				

```

1803      /SUBROUTINES: GETWORD AND GETCOMMAND1
1804      /SUBROUTINE TO GET AN EIGHT BIT WORD FROM THE INTERFACE.
1805      /IF TALKING TO A PDP8 INTERFACE IN 12 BIT MODE, THERE
1806      /WILL BE FOUR MEANINGLESS BITS PRECEDING THE DESIRED EIGHT
1807      /BIT WORD, ENTER THIS SUBROUTINE WITH THE RETURN ADDRESS
1808      /IN THE COUNTER, EXIT WITH THE ONES COMPLIMENT OF THE
1809      /DESIRED WORD IN THE SHIFT REGISTER. PARITY IS COMPUTED AND
1810      /CHECKED ON ALL WORDS.
1811
1812
1813      2000 0012  GETWRD, SET XREQ      /REQUEST A WORD FROM INTERFACE
1814
1815      2001 0075  GETCMD, LSR          /STASH THE RETURN ADDRESS
1816      2002 0270      OPEN RTNA
1817      2003 0064      LSP
1818
1819      2004 2070      LCT              /CALL SUBR WAITRN TO WAIT FOR A WORD
1820      2005 1346      PGOTIT
1821      2006 0222      JUMP F4
1822      2007 2312      WAITRN
1823
1824      2010 0061  GOTIT, OFF FLAG      /CLEAR FLAG FOR PARITY CHECK
1825
1826      2011 0004      CLR ENR          /IN CASE RUN WAS A RESPONSE TO DONE
1827      2012 0020      CLR DONE
1828
1829      2013 0070      LCT              /SET UP BIT COUNT IN CNTR, 8 BIT OR 12 BIT
1830      2014 0367      -8-1
1831      2015 0150      BR XIIBIT F
1832      2016 2021      +3
1833      2017 0070      LCT
1834      2020 0363      -12-1
1835
1836      2021 0112  WATDAT, BR DATAIN ONE /WHAT IS THE DATA BIT?
1837      2022 2030  GOTONE              /ITS A ONE, GO SAVE IT
1838
1839      2023 0126      BR COFL T        /ITS A ZERO, WAS IT THE PARITY BIT (9TH BIT)?
1840      2024 2041  CHKPAR              /YES, GO CHECK PARITY
1841
1842      2025 0076      ROTATE ONE       /NO SAVE THE DATA BIT COMPLIMENTED IN SR
1843
1844      2026 0222      JUMP F4          /GO SHIFT UP ANOTHER BIT.
1845      2027 2034      NUTHER
1846
1847
1848
1849
1850      2030 2063  GOTONE, TUG FLAG     /COMPLIMENT THE PARITY GENERATOR
1851
1852      2031 0126      BR COFL T        /WAS IT THE PARITY BIT?
1853      2032 2041  CHKPAR              /YES, GO CHECK PARITY
1854
1855      2033 0074      ROTATE ZERO      /NO, SAVE THE COMPLIMENTED DATA BIT IN SR
1856
1857      2034 2026  NUTHER, SET SHIFT    /SHIFT PULSE AND INCREMENT BIT COUNT

```

```

1858      2035 0273      ICT
1859      2036 2024      CLR SHIFT
1860
1861      2037 0222      JUMP F4          /GO TEST THIS BIT.
1862      2040 2021  WATDAT
1863
1864
1865      2041 2176  CHKPAR, BR FLAGD ONE /WHERE THERE AN ODD NO. OF ONES?
1866      2042 2076  GOTWRD              /YES, PARITY WAS GOOD
1867
1868      2043 0214      OPEN STAT        /NO, STAT TO SR
1869      2044 0071      ESP
1870      2045 2075      LSR
1871
1872      2046 0070      LCT              /END AROUND SHIFT OF UPPER 5 BITS OF STAT IN SR
1873      2047 2072      -5-1
1874      2050 0122      BR SR7 T
1875      2051 2055      +4
1876      2052 0074      ROTATE ZERO
1877      2053 0222      JUMP F4
1878      2054 2056      +2
1879      2055 0076      ROTATE ONE
1880      2056 0073      ICT
1881      2057 0124      BR COFL F
1882      2060 2050      -8
1883
1884      2061 0074      ROTATE ZERO      /CLEAR INIT DONE
1885      2062 0076      ROTATE ONE      /SET PARITY ERROR
1886
1887      2063 0122      BR SR7 T        /END AROUND SHIFT OF CRC ERROR BIT OF STAT IN SR
1888      2064 2070      +4
1889      2065 0074      ROTATE ZERO
1890      2066 0222      JUMP F4
1891      2067 2071      +2
1892      2070 0076      ROTATE ONE
1893
1894      2071 2064      LSP              /RESTORE STAT TO SCRATCH PAD
1895
1896      2072 0070      LCT              /ERRCODE FOR PARITY ERROR
1897      2073 0210      KPARER
1898      2074 0226      JUMP F5
1899      2075 2010      GOERDN
1900
1901      2076 0270  GOTWRD, OPEN RTNA    /WORD WAS GOOD, EXIT FROM GETWRD, GETCMD
1902      2077 0203      JUMP F2 IND

```



```
1903
1904      /([SUBROUTINE: STEPHEAD])
1905      /THIS SUBROUTINE WILL STEP THE SPECIFIED NUMBER OF TRACKS IN THE
1906      /SPECIFIED DIRECTION. DIRECTION IS DETERMINED BY THE HD DIR FLOP
1907
1908      /THE NUMBER OF STEPS IS IN THE SR. RETURN ADDRESS IS IN THE CNTR.
1909      /EXIT IS TO THE RETURN ADDRESS IF HOMP IS DETECTED. EXIT IS TO RETURN
1910      /PLUS 2 IF THE LAST STEP HAS BEEN TAKEN. AFTER THE LAST STEP IS TAKEN,
1911      /THE HEAD IS LOADED AND A 25MS DELAY IS EXECUTED FOR HEAD SETTLE TIME
1912
1913
1914
1915
1916      2100 0270      STEPHD, OPEN RTNA      /STORE RETURN ADDR AND MOVE STEP COUNT TO CNTR
1917      2101 0064      LSP
1918      2102 0075      LSR
1919      2103 0071      ESP
1920      2104 0064      LSP
1921
1922      2105 0136      CKHOME, BR HOME T      /IS THE HEAD HOME?
1923      2106 2150      OUT                    /YES, GO EXIT
1924
1925      2107 0073      ICT                    /NO, INCREMENT STEP COUNT AND STORE IN TEMPA
1926      2110 0075      LSR
1927      2111 0230      OPEN TEMPA
1928      2112 0064      LSP
1929
1930      2113 0270      LCT                    /PASS 30 TO DELAY SUBR FOR 3MS DELAY
1931      2114 2124      SECPLS
1932      2115 0075      LSR
1933      2116 0070      LCT
1934      2117 0341      -30-1
1935
1936      2120 0012      SET STPHD              /ISSUE STEP PULSE
1937      2121 0010      CLR STPHD
1938
1939      2122 0212      JUMP F2                /CALL DELAY SUBR
1940      2123 1300      DELAY
1941
1942      2124 0212      SECPLS, SET STPHD      /ISSUE SECOND STEP PULSE
1943      2125 0010      CLR STPHD
1944
1945      2126 0270      LCT                    /CALL DELAY FOR 3MS DELAY
1946      2127 2135      DONSTP
1947      2130 0075      LSR
1948      2131 0270      LCT
1949      2132 0341      -30-1
1950      2133 0212      JUMP F2
1951      2134 1300      DELAY
1952
1953      2135 0230      DONSTP, OPEN TEMPA     /CHECK STEP COUNT
1954      2136 0071      ESP
1955      2137 2124      BR COFL F
1956      2140 2125      CKHOME                /NOT DONE, GO CHECK IF HOME
1957
```

```
1958      2141 0270      OPEN RTNA          /DONE STEPPING, INCREMENT RETURN ADDRESS BY 2
1959      2142 0071      ESP
1960      2143 0073      ICT
1961      2144 0073      ICT
1962
1963      2145 0270      DLY25, OPEN RTNA      /STORE RETURN ADDRESS ALSO START OF 25MS DELAY SUBROUTINE
1964      2146 0075      LSR
1965      2147 0064      LSP
1966
1967      2150 0042      OUT, LDHD              /LOAD HEAD
1968      2151 0250      OPEN TEMPE          /SET SOFT HD LOAD BIT
1969      2152 0070      LCT
1970      OCTAL
1971      2153 0200      200
1972      DECIMAL
1973      2154 0075      LSR
1974      2155 0064      LSP
1975
1976      2156 0270      LCT                    /CALL DELAY SUBR FOR 25MS DELAY
1977      2157 2165      DONDLY
1978      2160 0075      LSR
1979      2161 0270      LCT
1980      2162 0000      -255-1
1981      2163 0212      JUMP F2
1982      2164 1300      DELAY
1983
1984      2165 0270      DONDLY, OPEN RTNA      /RETURN FROM STEP HEAD OR DELAY 25MS SUBROUTINE
1985      2166 0203      JUMP F0 IND
1986
1987
1988
1989
1990
1991
1992
1993      /([ROUTINE: READ SECTOR CONT.])
1994
1995      2167 0070      READ, LCT              /3 TO DATA MARK TRY COUNTER
1996      2170 0374      -3-1
1997      2171 0234      OPEN TEMPB
1998      2172 0075      LSR
1999      2173 0064      LSP
2000
2001      2174 0070      LCT                    /STALL FOR 96 MICRO SEC (3 BYTES) TO AVOID WRT TURN ON SPLASH
2002      2175 0207      -120-1
2003      2176 0073      ICT
2004      2177 0062      FLAG ON              /SET THE FLAG TO SPECIFY DATA AM IN FIND AM ROUTINE
2005      2200 0124      BR COFL F
2006      2201 2176      -3
2007
2008      2202 0073      ICT                    /CLR COUNTER AND SR
2009      2203 0075      LSR
2010
2011      2204 0216      JUMP F3
2012      2205 1441      GETDAM
```

36

```

2013
2014
2015      /THIS ROUTINE FOLLOWS THE DISCOVERY OF A DATA MARK OR
2016      /A DELETED DATA MARK. IT MOVES THE NEXT 1024 BITS
2017      /INTO THE SECTOR BUFFER, THEN PICKS UP AND CHECKS THE CRC.
2018
2019      2206 0044      DATA, CLR BAR      /CLEAR THE BUFFER ADDRESS REGISTER
2020
2021      2207 0144      BR SEPCLK F      /WAIT FOR CLOCK
2022      2210 2207      .-1
2023
2024      2211 0053      START WRTBUF      /START THE WRITE PULSE FOR THIS BIT
2025
2026      2212 0055      DATCRC      /UPDATE THE CRC WITH SEP DATA
2027
2028      2213 0162      BR BAROFL T      /IS BUFFER FULL YET?
2029      2214 2221      GETCRC      /YES, GO GET THE CRC
2030
2031      2215 0050      FIN WRTBUF      /NO, END THE WRITE PULSE
2032
2033      2216 0046      INCR BAR      /ADDRESS NEXT SECTOR BUFFER CELL
2034
2035      2217 0222      JUMP F4      /LOOP BACK FOR NEXT BIT
2036      2220 2207      DATA+1
2037
2038      2221 0050      GETCRC, FIN WRTBUF      /END THE WRITE PULSE FOR THE LAST BIT
2039
2040      2222 0070      LCT      /SET BIT COUNT TO 16 FOR 2 BYTE CRC
2041      2223 0357      -16-1
2042
2043      2224 0144      BR SEPCLK F      /WAIT FOR NEXT BIT
2044      2225 2224      .-1
2045
2046      2226 0042      LDMD      /4 NOOPS FOR LONG SEP CLOCK
2047      2227 0042      LDMD
2048      2230 0042      LDMD
2049      2231 0042      LDMD
2050
2051      2232 0055      DATCRC      /PUT CRC BIT IN THE CRC GENERATOR
2052      2233 0073      ICT      /INCREMENT AND TEST BIT COUNT
2053      2234 0124      BR COFL F
2054      2235 2224      .-9      /NOT DONE, GET ANOTHER
2055
2056      2236 0214      OPEN STAT      /STATUS TO SHIFT REG
2057      2237 0071      RSP
2058      2240 0375      LSR
2059
2060      2241 0122      BR SP7 T      /END AROUND SHIFT OF DRV RDY BIT OF STAT IN SR
2061      2242 2246      .+4
2062      2243 0074      ROTATE ZERO
2063      2244 0222      JUMP F4
2064      2245 2247      .+2
2065      2246 0076      ROTATE ONE
2066
2067      2247 0176      BR FLAG0 T      /SET DEL DATA BIT OF STAT IF FLAG=0

```

```

2068      2250 2254      .+4
2069      2251 0276      ROTATE ONE
2070      2252 0222      JUMP F4
2071      2253 2255      .+2
2072      2254 0074      ROTATE ZERO
2073
2074      2255 0070      LCT      /END AROUND SHIFT OF NEXT 5 BITS OF STAT IN SR
2075      2256 0372      -5-1
2076      2257 0122      BR SP7 T
2077      2260 2264      .+4
2078      2261 0074      ROTATE ZERO
2079      2262 0222      JUMP F4
2080      2263 2265      .+2
2081      2264 0074      ROTATE ONE
2082      2265 0073      ICT
2083      2266 0124      BR COFL F
2084      2267 2257      .-8
2085
2086      2270 0070      LCT      /SET BIT COUNTED TO 16 FOR CRC TEST
2087      2271 0357      -16-1
2088
2089      2272 0132      BR CRC16 ONE      /IS THIS CRC BIT OK
2090      2273 2304      DCR CER      /NO, GO REPORT DATA CRC ERROR
2091
2092      2274 0054      CRC ZERO      /YES, BRING UP NEXT CRC BIT
2093
2094      2275 0073      ICT      /INCREMENT AND TEST BIT COUNTER
2095      2276 0124      BR COFL F
2096      2277 2272      .-5      /GO CHECK ANOTHER
2097
2098      2300 0074      ROTATE ZERO      /CRC WAS GOOD, CLR CRC ERR BIT OF STAT IN SR
2099
2100      2301 0064      LSP      /PUT THE STATUS WORD BACK IN SCRATCHPAD
2101
2102      2302 0212      JUMP F2      /EXIT TO DONE
2103      2303 1006      OKDONE
2104
2105      2304 0076      DCR CER, ROTATE ONE      /INSERT 1 INTO CRC ERROR BIT
2106
2107      2305 0064      LSP      /PUT THE STAT WORD BACK
2108
2109      2306 0070      LCT      /ERROR CODE FOR CRC ERROR
2110      2307 0200      KDCR CER
2111      2310 0226      JUMP F5
2112      2311 2610      GOERDN

```

```
2113      /([SUBROUTINE: WAIT FOR RUN])
2114      /THIS SUBROUTINE WILL WAIT FOR RUN. IF 46MS ELAPSES, THE HEAD IS UNLOADED
2115      /AND THE ROUTINE CONTINUES WAITING FOR RUN. RETURN ADDRESS IS PASSED
2116      /VIA THE COUNTER
2117
2118
2119      2312 0264 WAITRN, OPEN RTNB      /STASH THE RETURN ADDRESS
2120      2313 0075 LSR
2121      2314 0264 LSP
2122
2123      2315 0102 BR RUN T      /GOT RUN?
2124      2316 2347 GOTRUN
2125
2126      2317 0240 OPEN TEMPC      /PRESET LOOP COUNTER TO 0
2127      2320 0070 LCT
2128      2321 0000 0
2129
2130      2322 0075 BACK, LSR      /RESTORE LOOP COUNT
2131      2323 0064 LSP
2132
2133      2324 0302 WBR RUN T      /TIME WHILE WAITING FOR RUN
2134      2325 2347 GOTRUN
2135      2326 0302 WBR RUN T
2136      2327 2347 GOTRUN
2137      2330 0302 WBR RUN T
2138      2331 2347 GOTRUN
2139      2332 0302 WBR RUN T
2140      2333 2347 GOTRUN
2141
2142      2334 0071 ESP      /INCREMENT AND TEST LOOP COUNT
2143      2335 0073 ICT
2144      2336 0124 BR COPL F
2145      2337 2322 BACK      /46MS NOT ELAPSED YET
2146
2147      2340 0250 OPEN TEMPE      /TIME IS EXPIRED (45.8 MS). CLEAR THE SOFT HOLD BIT AND UNLOAD THE HEAD
2148      2341 0073 ICT
2149      2342 0075 LSR
2150      2343 0064 LSP
2151      2344 0040 UNHD
2152
2153      2345 0100 BR RUN F      /WAIT FOR RUN. FOREVER IF NECESSARY
2154      2346 2345 .-1
2155
2156      2347 0010 GOTRUN, CLR XREQ      /IF RUN WAS RESPONSE TO XFREQ
2157
2158      2350 0264 OPEN RTNB      /RETURN FROM WAITRN SUBROUTINE
2159      2351 0213 JUMP IND F2
2160
2161
```

```
2162      /([ROUTINE: INITIALIZE CONT.])
2163      /CONTINUATION OF THE INITIALIZE SELF TEST
2164
2165      2352 0070 TEST, LCT      /LOAD R5 WITH TEST PATTERN 252
2166      OCTAL
2167      2353 0252 252
2168      DECIMAL
2169      2354 0075 LSR
2170      2355 0224 OPEN R5
2171      2356 0064 LSP
2172
2173      2357 0070 LCT      /LOAD R10 WITH TEST PATTERN 125
2174      OCTAL
2175      2360 0125 125
2176      DECIMAL
2177      2361 0075 LSR
2178      2362 0250 OPEN R10
2179      2363 0064 LSP
2180
2181      2364 0062 FLAG ON      /SET FLAG AND TEST IT
2182      2365 0176 BR FLAGO T
2183      2366 2371 .+3
2184      2367 0212 JUMP F2
2185      2370 1374 INTER1      /FLAG FAILURE
2186
2187      2371 0224 OPEN R5      /CONTENTS OF R5 TO SR, SHOULD BE 252
2188      2372 0071 ESP
2189      2373 0075 LSR
2190
2191      2374 0212 JUMP F2      /GO CONTINUE INIT TEST IN FLD 2
2192      2375 1351 TEST1
2193
2194
2195      2376 0000 0      /OPEN
2196      2377 0000 0      /OPEN
```

```

2197      /([SUBROUTINE: MAGNITUDE COMPARISON])
2198      /THIS SUBROUTINE COMPARES THE EIGHT BIT NUMBERS IN REGISTERS F AND G
2199      / EXIT IS TO THE RETURN ADDRESS IF F=G. IF F<G, RETURN IS TO RTNA+2.
2200      /IF F>G, RETURN IS TO RTNA+4. CONTENTS OF F AND G ARE UNDEFINED AT
2201      /THE END OF THE SUBROUTINE
2202
2203
2204
2205
2206      2400 0230      MAGCOM, OPEN TEMPA      /FOR BIT COUNT
2207
2208      2401 0070      LCT      /BIT COUNT IS 8
2209      2402 0367      -8-1
2210
2211      2403 0075      LSP      /RESTORE BIT COUNT
2212      2404 0064      LSP
2213
2214      2405 0254      OPEN TEMPF      /F TO SR
2215      2406 0071      ESP
2216      2407 0075      LSR
2217
2218      2410 0120      BR SR7 ZERO      /TEST F
2219      2411 2443      TSTG0      /ITS 0
2220
2221      2412 0076      ROTATE ONE      /ITS 1, BRING UP NEXT BIT
2222
2223      2413 0064      LSP      /RESTORE F
2224
2225      2414 0260      OPEN TEMPG      /G TO SR
2226      2415 0071      ESP
2227      2416 0075      LSR
2228
2229      2417 0120      BR SR7 ZERO      /TEST G
2230      2420 2432      GLESSF      /ITS 0, G IS LESS THAN F
2231
2232      2421 0074      NEXTG, ROTATE ZERO      /ITS 1, BRING UP NEXT G BIT
2233
2234      2422 0064      LSP      /RESTORE G
2235
2236      2423 0230      OPEN TEMPA      /INCREMENT AND TEST BIT COUNT
2237      2424 0071      ESP
2238      2425 0073      ICT
2239      2426 0124      BR COFL F
2240      2427 2403      MAGCOM+3      /GO COMPARE ANOTHER BIT
2241
2242      2430 2272      OPEN RTNA      /ALL BITS COMPARED, NO DIFFERENCE
2243      2431 0203      JUMP F0 IND
2244
2245      2432 0270      GLESSF, OPEN RTNA      /G IS LESS THAN F RETURN TO RTNA +4
2246      2433 0071      ESP
2247      2434 0073      ICT
2248      2435 0073      ICT
2249      2436 0073      ICT
2250      2437 0073      ICT
2251      2440 0075      LSR

```

```

2252      2441 0064      LSP
2253      2442 0203      JUMP F0 IND
2254
2255      2443 0074      TSTG0, ROTATE ZERO      /F HAS 0, BRING UP NEXT BIT
2256
2257      2444 0064      LSP      /RESTORE F
2258
2259      2445 0260      OPEN TEMPG      /G TO SR
2260      2446 0071      ESP
2261      2447 0075      LSR
2262
2263      2450 0120      BR SR7 ZERO      /TEST G
2264      2451 2421      NEXTG      /MATCHES F, GO BRING UP NEXT G BIT
2265
2266      2452 0270      OPEN RTNA      /G IS LESS THAN F, RETURN TO RTNA +2
2267      2453 0071      ESP
2268      2454 0226      JUMP F5
2269      2455 2436      GLESSF+4
2270
2271
2272
2273      /([SUBROUTINE: FIND TRACK CONT.])
2274
2275      2456 0270      HOMERR, LCT      /HOME FOUND BEFORE LAST STEP TAKEN
2276      2457 0050      KHOMERR
2277      2460 0226      JUMP F5
2278      2461 2610      GOERDN
2279
2280
2281      /([SUBROUTINE: DIFFERENCE])
2282      /THIS SUBROUTINE COMPUTES THE DIFFERENCE BETWEEN TWO EIGHT BIT
2283      /NUMBERS. ENTER WITH THE RETURN ADDRESS IN RTN, A IN THE
2284      /COUNTER AND B IN THE SHIFT REGISTER. EXIT IS MADE WITH THE
2285      /COMPLIMENT OF THE DIFFERENCE IN THE SHIFT REGISTER.
2286      /EXIT IS TO RTN IF A>=B. EXIT IS TO RTN+2 IF A<B
2287
2288
2289
2290
2291      2462 0230      DIF, OPEN TEMPA      /OPEN TEMPORARY PATH THRU THE SP
2292
2293      2463 0120      BR COFL T      /HAS A REACHED ALL ONES YET?
2294      2464 2501      DIFB      /YES, GO GET A FOR THE DIFFERENCE
2295
2296      2465 0064      LSP      /NO, GET B
2297      2466 0075      LSR      /A INTO SHIFT REG
2298      2467 0071      ESP      /B INTO COUNTER
2299
2300      2470 0120      BR COFL T      /HAS B REACHED ALL ONES YET?
2301      2471 2503      DIFA      /YES, GO GET A FOR THE DIFFERENCE
2302
2303      2472 0073      ICT      /INCREMENT B
2304      2473 0064      LSP      /, BRING BACK A
2305      2474 0075      LSR      /B INTO SHIFT REG
2306      2475 0071      ESP      /A INTO COUNTER

```

2307	2476	0073	ICT	/INCREMENT A
2308	2477	0226	JUMP FS	/GO BACK TO TEST A AGAIN
2309	2502	2463	DIF+1	
2310				
2311				
2312	2501	0270	DIFB, OPEN RTNA	/H IS THE COMPLIMENT OF THE DIFFERENCE
2313	2502	0203	JUMP F0 IND	/EXIT A>=B
2314				
2315				
2316	2503	0270	DIFA, OPEN RTNA	/A IS THE COMPLIMENT OF THE DIFFERENCE
2317	2504	0071	ESP	/INCREMENT THE RETURN ADDRESS BY 2
2318	2505	0073	ICT	
2319	2506	0073	ICT	
2320				
2321	2507	0064	LSP	/RESTORE RETURN ADDRESS TO SCRATCHPAD AND 1 TO SR
2322	2510	0075	LSR	
2323	2511	0071	ESP	
2324	2512	0064	LSP	
2325	2513	0075	LSR	
2326				
2327	2514	0203	JUMP F0 IND	/EXIT A<B
2328				
2329				
2330				
2331				
2332				
2333				
2334				
2335				
2336				
2337				
2338	2515	0070	CKHCRC, LCT	/PRESET BIT COUNT TO 16 FOR CRC
2339	2516	0357	-16-1	
2340				
2341	2517	0132	BR CRC16 ONE	/IS CRC ZERO
2342	2520	2546	HCRCER	/NO, LOG ERROR AND TRY AGAIN
2343				
2344	2521	0073	ICT	/YES, CRC GOOD SO FAR, BUMP BIT CNTR
2345				
2346	2522	0254	CRC ZERO	/BRING UP NEXT CRC BIT
2347				
2348	2523	0124	BR COFL F	/ALL BITS TESTED?
2349	2524	2517	.-5	/NO, BRANCH BACK
2350				
2351	2525	0210	OPEN ERREG	/YES, CRC WAS GOOD, CHECK TRK COMP
2352	2526	0271	ESP	
2353	2527	0075	LSR	
2354				
2355	2530	0270	LCT	/ROTATE BIT 0 TO BIT 7
2356	2531	0370	-7-1	
2357	2532	0074	ROTATE ZERO	
2358	2533	0073	ICT	
2359	2534	0124	BR COFL F	/DONE ROTATING?
2360	2535	2532	.-3	/NO
2361				

2362	2536	0122	BR SR7 ONE	/YES, WAS THERE A BAD COMPARE
2363	2537	2542	TKSKER	/YES, GO REPORT A TRACK SEEK ERROR
2364				
2365	2540	0264	OPEN RTNA	/CORRECT TRACK, EXIT FROM FIND HDR SUBR
2366	2541	0207	JUMP F1 IND	
2367				
2368				
2369				
2370				
2371	2542	0070	TKSKER, LCT	/HEADER CRC WAS GOOD BUT TRACK
2372	2543	0150	TKSKER	/ADDRESS DID NOT COMPARE, MUST
2373	2544	0226	JUMP FS	/EXIT TO ERROR DONE
2374	2545	2610	GOERDN	
2375				
2376				
2377	2546	0070	HCRCER, LCT	/HEADER CRC WAS NOT CORRECT
2378	2547	0140	HCRCER	
2379	2550	0275	LSR	
2380	2551	0210	OPEN ERREG	/LOG THE ERROR
2381	2552	0064	LSR	
2382				
2383	2553	0226	JUMP FS	/GO TRY ANOTHER HEADER
2384	2554	2557	BADHDR	
2385				
2386				
2387				
2388				
2389	2555	0170	BDSRT, BR FLAGO T	/BAD START ON DATA AM OR IDAM?
2390	2556	2577	BADDAM	
2391				
2392	2557	0230	BADHDR, OPEN TEMPB	/IDAM, INCREMENT AND TEST BAD START INNER COUNT
2393	2560	0071	ESP	
2394	2561	0073	ICT	
2395	2562	0275	LSR	
2396	2563	0064	LSP	
2397	2564	0124	BR COFL F	/NO OVERFLOW, GO TRY ANOTHER HEADER
2398	2565	2615	PTRYAG	/INCREMENT AND TEST BAD START OUTER COUNT
2399	2566	0234	OPEN TEMPB	
2400	2567	0071	ESP	
2401	2570	0073	ICT	
2402	2571	0124	BR COFL F	/NO OVERFLOW, GO TRY AGAIN
2403	2572	2615	PTRYAG	/TOO MANY TRIES FOR A HEADER
2404	2573	0070	XSTRYS, LCT	
2405	2574	0160	KXSTRYS	
2406	2575	0226	JUMP FS	
2407	2576	2610	GOERDN	
2408				
2409				
2410	2577	0234	BADDAM, OPEN TEMPB	/BAD START ON DATA AM, INCREMENT AND TEST BAD START COUNT
2411	2600	0071	ESP	
2412	2601	0073	ICT	
2413	2602	0075	LSR	
2414	2603	0064	LSP	
2415	2604	0124	BR COFL F	/NO OVERFLOW GO TRY FOR DATA AM AGAIN
2416	2605	2617	PGETDA	

2417	2606	0070	NODAM, LCT	/TRIED 3 TIMES FOR DATA AM, GO FLAG THE ERROR
2418	2607	0170	KNODAM	
2419	2610	0210	GOERON, OPEN ERREG	
2420	2611	0075	LSR	
2421	2612	0064	LSP	
2422	2613	0212	JUMP F2	
2423	2614	1000	ENDONE	
2424				
2425	2615	0216	PTRYAG, JUMP F3	/POINTER TO FIND AN IDAM
2426	2616	1413	TRYAGN	
2427				
2428				
2429	2617	0216	PGETDA, JUMP F3	/POINTER TO FIND DATA AM
2430	2620	1441	GETDAM	
2431				
2432				
2433				
2434			/[ROUTINE: INITIALIZE CONT.]	
2435	2621	0070	WRONG, LCT	/HOME WAS FOUND WHILE STEPPING OUT
2436	2622	0030	KARONG	
2437	2623	0226	JUMP F5	
2438	2624	2610	GOERON	
2439				
2440	2625	0070	DNRCAL, LCT	/CALL CHECK READY SUBROUTINE
2441	2626	1771	PNORDY	
2442	2627	0226	JUMP F5	
2443	2630	2640	CHKRDY	
2444				
2445	2631	0070	INTRDY, LCT	/DRV 0 IS READY CALL BOOT SUBROUTINE TO
2446	2632	0770	GOREAD	/MOVE TO TRACK 1, THEN GO TO READ ROUTINE TO
2447	2633	0274	OPEN RTN	/PICK UP SECTOR 1
2448	2634	0075	LSR	
2449	2635	0064	LSP	
2450	2636	0202	JUMP F0	
2451	2637	0252	BOOT	
2452				
2453				
2454				
2455				

2456			/[SUBROUTINE: CHECKREADY]	
2457				
2458			/SUBROUTINE TO CHECK THE SELECTED DRIVE TO SEE IF THE	
2459			/DISK IS INSERTED AND UP TO SPEED. THIS IS DONE BY CHECKING TO SEE IF	
2460			/THE INTERVAL BETWEEN 2 INDEX PULSES IS BETWEEN 150 MS AND 180 MS. RETURN	
2461			/ADDRESS IS PLACED IN THE COUNTER BEFORE ENTRY. NOT READY RETURN IS	
2462			/TO THE RETURN ADDRESS. READY RETURN IS TO THE RETURN ADDRESS PLUS 2	
2463				
2464				
2465				
2466			/	
2467	2642	0274	CHKRDY, OPEN RTN	/SAVE RETURN ADDRESS
2468	2641	0075	LSR	
2469	2642	0064	LSP	
2470				
2471	2643	0070	LCT	/2 TO CNTR FOR INDEX PASS COUNT
2472	2644	0375	-2-1	
2473				
2474	2645	0230	OPEN TEMPA	/FOR INDEX PASS COUNT
2475				
2476	2646	0075	NEWPAS, LSR	/RESTORE INDEX PASS COUNT
2477	2647	0064	LSP	
2478				
2479	2650	0061	FLAG OFF	/CLOSE INDEX WINDOW
2480				
2481	2651	0042	LDMO	/TO CLEAR INDEX FLOP
2482				
2483	2652	0070	LCT	/FOR 15 TIMES THROUGH 10MS LOOP
2484	2653	0360	-15-1	
2485				
2486	2654	0234	STDLY, OPEN TEMPB	/RESTORE OUTER COUNT
2487	2655	0075	LSR	
2488	2656	0064	LSP	
2489				
2490	2657	0070	LCT	/FOR 40 TIMES THROUGH .25MS LOOP
2491	2660	0327	-40-1	
2492				
2493	2661	0240	OPEN TEMPC	/RESTORE INNER COUNT
2494	2662	0075	LSR	
2495	2663	0064	LSP	
2496				
2497	2664	0070	LCT	/WAIT .25 MS FOR INDEX
2498	2665	0005	-250-1	
2499	2666	0116	BR INDX T	
2500	2667	2714	SAWIND	/FOUND INDEX
2501	2670	0073	ICT	
2502	2671	0124	BR COFL F	
2503	2672	2666	-4	
2504				
2505	2673	0240	OPEN TEMPC	/INCREMENT AND TEST INNER COUNT
2506	2674	0071	ESP	
2507	2675	0073	ICT	
2508	2676	0124	BR COFL F	
2509	2677	2662	SPBACK	
2510				