PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 3 : G11B 7/00 — A1

(11) International Publication Number: WO 82/ 01957

(43) International Publication Date: 10 June 1982 (10.06.82)

(21) International Application Number: PCT/US81/01569

(22) International Filing Date: 27 November 1981 (27.11.81)

(31) Priority Application Number: 210,847

(32) Priority Date: 26 November 1980 (26.11.80)

(33) Priority Country: US

(71)(72) Applicants and Inventors: ACKERMAN, Richard, C. [US/US]; 610 Hillside Road, Glenview, IL 60625 (US). HURLBUT, Donovan, H. [US/US]; Route 1, Whitewater, WI 53190 (US). JEWER, Alan, A. [US/US]; Route 3, Whitewater, WI 53190 (US).

(74) Agents: LEHMAN, Edward, A. et al.; Hill, Van Santen, Steadman, Chiara & Simpson, 70th Floor Sears Tower, Chicago, IL 60606 (US).

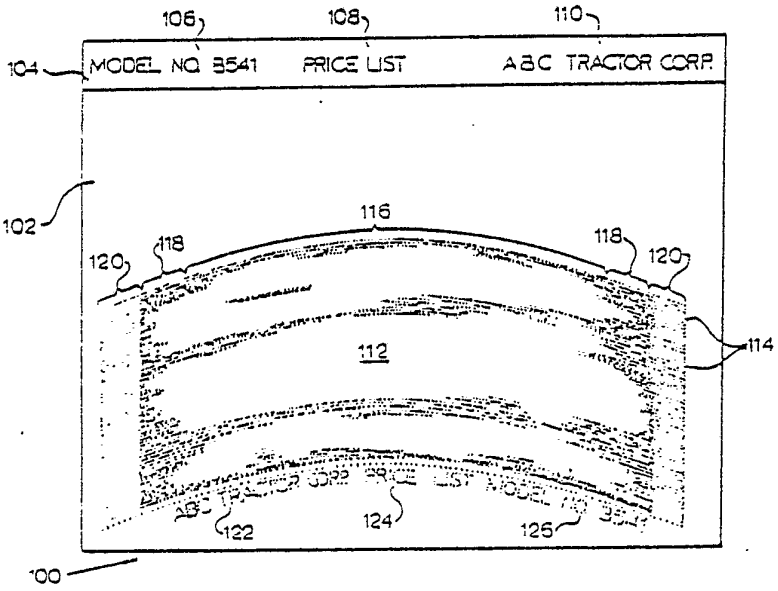
(81) Designated States: AT (European patent), AU, BR, CF (OAPI patent), CG (OAPI patent), CH (European patent), CM (OAPI patent), DE (European patent), DK, FR (European patent), GA (OAPI patent), GB (European patent), JP, LU (European patent), NL (European patent), NO, SE (European patent), SN (OAPI patent), SU, TD (OAPI patent), TG (OAPI patent).

Published
With international search report.

(54) Title: A DATA CARRIER AND APPARATUS FOR OPTICALLY READING DIGITAL DATA INSCRIBED IN AN ARCUATE PATTERN ON A DATA CARRIER

(57) Abstract

An information record (100) has a body (102) of photographic film, paper, plastics material or the like, in which digital information (112, 116, 118, 120) is carried on a body in equal-radii, arcuate rows (114) which extend across the body. The information may be stored as dark spots, light spots or dark-to-light or light-to-dark transitions with respect to transparent substances and in reflective areas, non-reflective areas or reflective-to-non-reflective or non-reflective-to-reflective transitions. The data are read with a beam produced by a laser (388, 434) upon relative movement between the laser (388, 434) and the data record (100). In one embodiment the laser (434) is fixed and the data record (100) is rotated, while in another embodiment the laser (388) is rotated and the data record (100) is relatively fixed, but linearly incrementally moved transversely of the beam path.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	KP	Democratic People's Republic of Korea
AU	Australia	LI	Liechtenstein
BR	Brazil	LU	Luxembourg
CF	Central African Republic	MC	Monaco
CG	Congo	MG	Madagascar
CH	Switzerland	MW	Malawi
CM	Cameroon	NL	Netherlands
DE	Germany, Federal Republic of	NO	Norway
DK	Denmark	RO	Romania
FI	Finland	SE	Sweden
FR	France	SN	Senegal
GA	Gabon	SU	Soviet Union
GB	United Kingdom	TD	Chad
HU	Hungary	TG	Togo
JP	Japan	US	United States of America

"A DATA CARRIER AND APPARATUS FOR OPTICALLY READING DIGITAL DATA INSCRIBED IN AN ARCUATE PATTERN ON A DATA CARRIER"

D E S C R I P T I O N

"A Data Carrier Method and Apparatus for Optically Reading Digital Data Inscribed in an Arcuate Pattern on a Data Carrier"

The present invention relates to a method and apparatus for optically reading data from an information record or information carrier in which the data is inscribed in an arcuate pattern. More particularly, the invention concerns the retrieval of digital information from a data carrier in which the digital information is stored on a body in parallel, spaced-apart rows, the rows extending in an arcuate manner across the body.

Gokey et al U.S. Patent 4,213,040 and Gokey et al U.S.Serial No. 956,426 disclose the provision of digital information in rows and columns on a data carrier, in which access and reading is performed with an X-Y transport adapted for skew correction.

Gokey et al U.S. Serial No. 074,095 provides an information record which takes the form of a disc in which rows of information are disposed as straight radial lines, arcuate radial lines or circles with respect to the axis of rotation of the disc.

Ackerman et al Serial No. 181,172 discloses an information record of the type which may be used in practicing the present invention and, in fact, forms the accompanying FIG. 1. and is included in this invention.

The object of the present invention is to provide a method and apparatus for optically retrieving digital information from a data carrier in which the advantages of the rectangular carrier (skew correction and the like) is maintained along with the advantages of a constantly rotating and constantly scanning system, while avoiding the disadvantages associated with both types of systems.

According to the invention, digital information is read from a body of material, such as photographic film, paper, various plastics material and the like, in which the data is stored in spaced-apart, arcuate rows. The data is scanned by a moving light beam, preferably generated by a laser source, so that the light beam is modulated by the reflective-to-nonreflective character of the data on the data carrier, and as may be augmented by a reflective support behind the carrier, depending upon the material used for the data carrier. The material may vary, depending upon application, and may include paper, coated enamel paper, plastic filament paper, Mylar, Kodalith Pan, Dry Silver, Tri-X, Plus X, Diazo and Vesicular materials.

In one embodiment of apparatus for practicing the invention, the light source is fixed and is guided by mirrors, prisms and the like for emission from a wheel which rotates past the data carrier.

In another embodiment, a preferred embodiment, the light source is mounted on a rotating wheel and the light beam is directed in a similar manner so as to scan the data carrier. Inasmuch as the light path in this embodiment is much shorter than that of the fixed light source type of system, a much greater efficiency is achieved.

According to the invention, the method includes generating a beam of light, preferably a coherent beam of light, directing the beam of light so as to scan a data carrier along arcuate paths which correspond to the arcuate paths of data s age on the data carrier, reflecting the beam of light as modulated by the data and directing the same along a path, and receiving and converting the modulated light beam into electrical

signals which correspond to the data stored on the data carrier.

The apparatus for optically reading digital data which is inscribed in an arcuate pattern on a data carrier comprises a data carrier support for receiving and holding a data carrier, a light source operable to emit a beam of light, preferably coherent light, a beam directing means for receiving and directing the beam of light toward the supported data carrier, including means for sweeping the light beam across the data carrier in an arcuate path which corresponds to the arcuate storage pattern of the data on the carrier; and receiving means mounted to receive the light beam after the same strikes the data carrier. As mentioned above, in one embodiment the light source is fixed and the light beam is directed to a beam sweeping structure, while in another embodiment, the light source is rotatably mounted with the beam directing means for causing the light beam to sweep across the data carrier.

ON THE DRAWINGS

Other objects, features and advantages of the invention, its organization, construction and operation will be best understood from the following detailed description, taken in conjunction with the accompanying drawings, on which:

FIG. 1 is a plan view of an information record of the type which may be employed in practicing the present invention; and is a part thereof;

FIG. 2 is an isometric view of an apparatus for practicing the present invention with the outer cover removed, but shown in phantom;

FIG. 3 is a sectional view taken substantially

along the parting line III-III of FIG. 2;

FIG. 4 is an isometric type of plan view of the base plate and apparatus supported by the base plate, including the data carrier support and carriage structure along with the skew correction and carriage incrementing motors;

FIG. 5 is an exploded view of the carriage structure of FIG. 4 and a wheel which mounts thereabove and carries the light source and light beam directing structure, and specifically illustrating a radius control motor to compensate for manufacturing tolerances, ambient temperature conditions and the like with respect to the size of the data carrier;

FIG. 6 is a schematic circuit diagram of the motor control circuits for controlling the skew and incrementing motors;

FIG. 7 is a schematic circuit diagram of a motor control circuit for controlling the radius motor;

FIGS. 8 and 9 are schematic illustrations of the incrementing or carriage drive motor and the skew motor;

FIG. 10 shows connector pin assignments for interconnecting the control circuits of FIG. 6 with the motors of FIGS. 8 and 9 and the control inputs effective to cause operation of these motors;

FIG. 11 is a schematic and sectional view of a structure for transferring the modulated light beam as electrical signals to a computer;

FIG. 12 is a schematic representation of a fixed light source with a rotating beam directing system, as seen from the top; and

FIG. 13 is a portion of the system of FIG. 12 as seen from the right-hand side.

Before turning to the method and apparatus for

reading a data record, reference should be first made to FIG. 1 which illustrates a data carrier for use in practicing the present invention. The data carrier is generally illustrated at 100 as comprising a body 102 which may have a header 104 affixed thereto. The header 104 may include visually perceptible information, as illustrated at 106, 108 and 110, dealing with, for example, the price list relating to a particular model number manufactured by a particular manufacturing organization. This information could also relate to any type of inventory and may find particular use in the banking, hardware, hospital and pharmaceutical areas.

As a particular example, the digital information relating to each element which may be a part of a price list, for example, is recorded in a plurality of parallel-appearing, spaced-apart rows of information 114 in an area 112 of the body 102.

Although an operating program may tell a machine the address of a row with respect to distance from a datum line, so that the machine itself will respond accordingly to access the proper row, each row may include a row address at either or both ends, as referenced at 118, with the digital information to be retrieved located therebetween, such as at 116. The address information 118 may also be interspersed along the respective arcuate data line, assuming that the hardware and software is set up to read in such a manner.

The record medium may be in the form of paper, plastics or the like, or may be a photographic film in which the background may be transparent or dark and the individual information dots may be of opposite character. The dots themselves may contain the digital information such that a dark dot on a light or transparent background

may be considered a "1" or such that a light dot on a dark background may represent a "1". Preferably, however, and notwithstanding the transparency of the background, the information is contained in the opaque-to-transparent and transparent-to-opaque transitions. The transparency and opaqueness of a medium, for example a photographic film, is not limiting with respect to practicing the present invention. The information might be stored with respect to reflective elements carried by the body, whether the body is transparent or opaque. Again, the information may be stored as a result of the reflective material being present or absent, or may be stored with respect to the transitions between the reflective and non-reflective areas.

The camera for producing such a data record also has the capability of providing, in place of or in addition to the information 106, 108 and 110 carried on a header 104, the same or additional information on the film, as indicated at 122, 124 and 126, in the same manner as recording of the digital information. Such information may be graphic and/or digital, where graphic means all types of information, including alpha-numeric, drawings, symbols and the like.

The data record may also comprise leader (trailer) indicia 120, which informs the reading device of the beginning and ending of digital information.

Referring to FIGS. 2 and 3, the frame of a preferred apparatus for practicing the present invention is generally illustrated at 128 as comprising a top plate 130 and a base plate 132 which are fixed generally parallel to one another, spaced-apart, by way of a plurality of spacers 134. A front plate 136 is also provided and extends between the top plate 130 and the

base plate 132.

The top plate 130 includes an aperture 131 with an access plate 133 and carries, towards the rear thereof, a power supply 138 and a computer 140. The entire structure may be covered with a protective cover 142 which includes a slot 144 for alignment with a slot 146 in the front plate 136 to receive the data record 100 therethrough. The cover 142 may also include an aperture 148 for receiving the operating member of an on-off switch 150.

The on-off switch 150 controls the application of power to the entire unit by way of a line cord and plug assembly 152 and transformer 154, the line cord being received through a mounting bracket 156 at the left-hand, rear corner of the unit.

The power cord 152 may terminate a terminal block 158 which services the power supply 138 by way of a fuse holder 160 and further connections (not shown).

A motor 162 is mounted at the left-hand, rear corner of the unit by way of a mounting bracket 164, and receives power from the terminal block 158 by way of a connector 166 and, possibly, via a motor on-off switch 168.

The motor 162, upon energization, constantly rotates a wheel or disc 174 by way of a belt 170 which is received in a groove 172 of the wheel 174' carried by the wheel 174.

Referring specifically to FIG. 3, a shaft 176 is rotatably mounted between the top plate 130 and the base plate 132. The shaft 176 is mounted in an upper bearing 178 received in a plate 180 which is affixed to the top plate 130 by way of screws 182. In a similar manner, a lower bearing 184 receives the shaft 176 and is

mounted in a plate 186 which is affixed to the base plate 132 by way of screws 188.

The upper portion of the shaft carries a first slip ring assembly, while the lower portion of the shaft carries a second slip ring assembly. The upper, first slip ring assembly comprises a plurality of slip rings 200, 202, 204 which are insulated with respect to the shaft, while the lower slip ring assembly comprises a pair of slip rings 206 and 208 which are also insulated from the shaft. The upper slip ring assembly comprises respective spring-loaded contacts (brushes) 210, 212, 214, while the lower slip ring assembly comprises a pair of spring-loaded contacts 216 and 218 (brushes). The upper brushes 210, 212, 214 are mounted in an insulating brush holder 220 which is secured to the plate 180 by way of a screw or screws 222 and a pair of plates 224 and 226. In the same manner, the lower contacts are held in a brush holder 228 which is secured to the plate 186 by way of a screw or screws 230 and a pair of plates 232, 234. As seen in FIG. 5, a connector and cable assembly 236 is provided to feed voltage potentials to the upper contacts 210, 212, 214; a similar arrangement being provided, but not shown, for the lower contacts 216 and 218. The contacts 200, 210 may carry, for example, $\underline{+12}$ VDC; the contacts 202, 212 may carry ground; the contacts 204, 214 may carry $\underline{+}$ 19 VDC unregulated for the radius motor; and the contacts 206, 216 and 208, 218 may carry, for example, 700 VAC for the laser.

The wheel 174 includes a central aperture 190 which receives a portion of a stepped flange 192 and is secured thereto by a plurality of screws 194.

As shown and described, the motor 162 by way of the belt 170 rotates the wheel 174 between the top plate

130 and the bottom plate 132.

In reading data from the data record 100, the computer 140 must know at which time the data will be scanned, that is the computer must know the position of the wheel 174, at least immediately prior to data scanning. For this purpose, and as shown in a cut-away portion of the plate 130 in FIG. 2, a bracket 238 mounts a Hall-effect device and supporting circuit for generating electrical pulses, as clock pulses, in each instance as the wheel 174 carries a mounting bracket 244 mounting a magnet 242 there past. Therefore, for each revolution of the wheel 174, a pulse is delivered to the computer (connections not shown) for clocking data retrieval with respect to rotation.

Referring to FIG. 2, and as also illustrated in FIG. 4, the base plate 132 is provided with a threaded magnifier mounting ring 246. The mounting ring 246, when the unit is placed on its side, for example, receives a 50x magnifier for focusing. For this purpose, the wheel 174 is rotated such that a light beam via a prism 254 and a lens 258 (FIG. 3) are positioned above the mounting ring 246 and the magnifier. The lens 258 may therefore be adjusted to the focal plane of the data record 100, which plane is indicated in FIG. 3.

It should be pointed out here that the wheel 174 carries a plate 248 which mounts a prism holder 250 for holding a prism 254, the prism holder 250 receiving a lens mount 256 having a lens 258 therein. As the light beam, as indicated by broken lines, passes through the lens and is reflected from the media, the same is received, at a slight angle, for example 15°, by a photo detector 260, such as a photodiode or a phototransistor.

Turning now to FIGS. 4 and 5, the carriage

structure is illustrated in detail as comprising a carrier 264 which is mounted for incremental movement along a track 262 as supported by a plurality of guide wheel assemblies 268 and a guide wheel assembly 290. Each guide wheel assembly 268 comprises a guide wheel 270, a spacer 272, an eccentric bushing 274 and a screw 76. These elements are all on the right-hand side of the unit. On the left-hand side of the unit, supporting the carriage, is a wheel 292 which is adjusted in height by an eccentric bushing 294 and a screw 296 to level the carriage.

FIG. 5 illustrates that the guide wheel assemblies 268 are carried on respective flanges 278 and 280. On the rear flange 280, a magnet 298, via a washer 300 is secured in a magnet mount 320 which is, in turn, secured by way of screws 304 to the end of the flange 280. The magnet 298 cooperates with (FIG. 4) a Hall-effect device 308 and supporting circuitry 310 which is mounted adjacent the track 266 by way of a pair of circuitboard mounts 306. The movement of the magnet 298 toward and away from the Hall effect device 308 (connections not shown) tells the computer 140 whether the carriage 264 is in the home position or is away from the home position.

In order to increment the carriage arcuate line-by-arcuate line, a carriage motor 314 is mounted to the base plate 132 by way of a mounting bracket 312 and includes a lead screw-type output shaft which is rotatably connected to the carriage 264.

As will be understood from a more detailed description below, the carriage motor 314 is controlled by a motor control circuit 320 (detailed in FIG. 6) via a cable assembly 318.

The carriage structure also comprises a skew

pivot plate 322 which is mounted spaced from the carriage 264 by way of a plurality of spacers 324 and corresponding washers 328 and screws 326. The plate 322 includes a pivot hole 330, and a plurality of skew-accommodating slots 332, 334 and 336.

The skew plate 322 also includes a pair of L-shaped recesses 338 which open into a recess 340 in the top of the skew plate 322.

The recess 340 receives a glass or mirror plate 342 which is held down by a pair of small L-shaped springs 344 and respective screws 346. The ends 348 of a media holding spring 350 are received in the recesses 338 and secured therein by way of a pair of screws 352.

Referring still to FIGS. 4 and 5, the base plate 132 includes an elongate aperture 354 for receiving the body of a skew motor 356 for movement therethrough in accordance with the movements prescribed by the carriage motor 314. The skew motor 356 includes a front mounting bracket 358 which is connected to the rear end of a projection 360 of the carriage 264. The projection 360 includes a shaped slot 362 for receiving the forward (output shaft) portion 364 of the motor 356, the output shaft 364 contacting a lever 366 which is pivotally mounted at 368 to the carriage 264 by way of a shouldered portion 370 and a screw 372. The lever 366 includes an arm portion 374, and an adjustable screw 376, for contacting and pivoting the skew plate 322, while a skew return or bias spring 378 is connected between the skew pivot plate 322 and the motor mounting flange 358 (FIG. 4).

In reading data from a data record of the type illustrated in FIG. 1, care must be taken to accommodate the manufacturing tolerances and/or the temperature response of the record. For example, the accurate cutting

of the data record in the direction across the data, that is vertically in FIG. 1, changes the radius of rotation with respect to the radius of the arcuate pattern. Also, expansion or contraction of the data record due to ambient conditions changes this dimension. Therefore, apparatus is provided for accommodating radius changes due to manufacturing tolerances, ambient temperature and the like.

Referring to the upper portion of FIG. 5, radius correction structure is provided for the light beam emanating from the lens 258. This structure comprises a radius motor 380 which has an output lead screw 382 connected to the plate 248 at a pivot 384, the plate being pivotally mounted on the wheel 174 at a pivot 378 by means of a bushing, screw and the like (not shown). The motor 380 responds to control signals received from a radius motor control circuit 386, which is shown in detail in FIG. 7. As the disc rotates and the initializing operations are performed, the computer determines the direction of movement of the lens 258 and operates the radius motor 380 accordingly.

The light beam is generated, preferably, by a laser generator 388 which is powered by a respective power supply 390 fed from the master power supply 138. The laser generator 388 and the power supply 390 may advantageously be enclosed in a protective cover 392. The beam generated by the laser generator 388 is projected toward the periphery of the wheel 174 where it strikes a mirror 394 mounted on a mirror mount 396 and is reflected to the prism 254 for direction downwardly through the lens 258.

As illustrated in FIG. 3, the light beam through the lens 258 strikes the data record and is reflected

therefrom, modulated, to be received by the photo-detector 260. The photo detector 260 has output leads 398 which are connected to a circuit 400 which is an optical/optical converter to eliminate noise having an optical output 402 connected to an optical connector 404 which feeds a fiber optic 406. A representative circuit is illustrated in FIG. 14.

As illustrated in FIG. 3, the fiber optic 406 is received in a radial passageway 408 and extends up through an axial passageway 410 of the shaft 176 to an optical ferrule 112, whereupon the light emanating therefrom is detected by an optical detector 413 and fed to the computer 140 for buffering and the like.

A similar output may be had, referring to FIG. 11, by providing the output leads 398 to a circuit 414 which is essentially a pulse shaper having output leads 146 which extend through the radial passageway 408 and the axial passageway 410 of the shaft 176 to a light-emitting element 418, such as a light emitting diode. Again, a photo detector 420, such as a photo diode or a photo-transistor converts the light pulses to electrical pulses for feeding to the computer 144 for buffering and the like.

Referring to FIGS. 6--10, the motor control apparatus for the motors 314, 356 and 380 will be discussed.

The motor control circuitboard 320 for the skew and carriage motors is illustrated in FIG. 6 as a plurality of identical circuits 422 each comprising an optical isolator 424 and a switching transistor Q. The alphabet letters indicate computer inputs for receiving carriage incrementing and skew adjustment signals, while the numbered outputs indicate, in connection with FIGS.

8 and 9, the connections to the coils of the skew and carriage motors.

The radius motor 380 is similarly controlled, but with a dual circuit comprising optical isolators 426 and respective transistors Q9 and Q10 in order to feed one polarity or another to the skew control motor.

FIG. 10 relates the control inputs and the responsive outputs for the carriage, skew and radius motors with respect to the circuits of FIG. 6--9.

For those times during which it is necessary to tilt the unit on its side, for example during the focusing operation, a carriage retaining structure is provided for the left-hand side of the carriage. This structure comprises a L-shaped member 428 which includes a recessed surface 430 (from the bottom), secured to the base plate 132 by way of a pair of screws 432. As the carriage moves toward the front and toward the back, the left-hand edge thereof moves under the cut-out defining the surface 430 for retaining the carriage in the event of tilting for focusing, packing and the like.

Referring to FIGS. 12 and 13, FIG. 12 is a top view of an alternative optical system, while FIG. 13 is a portion of the optical system of FIG. 12 as seen from the right-hand side, inasmuch as the optical system of this embodiment covers at least three planes. In FIG. 12, and with a frame similar to that discussed above, a laser device 434 is fixed and emits a light beam 436 which is reflected to the right by a mirror 438 to form a beam segment 440 which, in turn, is reflected forwardly by a mirror 442 to form a beam segment 444. The beam segment 444 travels to a beam splitter 446 which passes a beam segment 448 to a prism 450 (shown as a mirror in FIG. 13 for simplicity) on the axis of the rotating wheel. The

prism 450 directs a beam segment 452 downwardly to a second prism 454, also on the axis of rotation which, in turn, directs a beam segment 456 toward a mirror 458 for reflection upwardly toward a data record 462 (100) which is movable incrementally toward the front and toward the back of the unit. The reflected light beam segment 464 again strikes the mirror 458 at a slight angle and is reflected as a beam segment 466 to the prism 454, where the same is redirected as a beam segment 468 to the prism 450. The prism 450 again reflects the modulated beam as a segment 464 to the beam splitter 446 where the modulated beam is reflected as a beam segment 472 to a photo detector 480 for conversion into an electrical signal and transmission to the computer 140 for buffering, output and the like.

It should be noted that in the preferred embodiment of FIGS. 2-5, as far as the mechanical rotating structure is concerned, and in the embodiment of FIG. 11, and in the embodiment of FIGS. 12 and 13, slip rings and the like for outputting the optical signal are not required, although the same could be utilized.

As another alternative, the laser and beam directing system could be generally fixed, but incremented, with respect to a rotating data record.

Also, instead of reflection of a modulated beam, the beam passage through a material having transparent portions could be detected and read with the beam passing through one side of the data record and being detected at the other side of the data record.

With the exception of specific fasteners (screws, washers, etc.) the following TABLE I is a schedule of components listed by reference character and source code. The following TABLE II identifies the sources with respect

to the source codes.

TABLE I

REF.NO.	COMPONENT	SOURCE	SOURCE STOCK NO.
130	Top Plate	A	
132	Base Plate	A	
133	Access Plate	A	
134	Support Column (5)	A	
136	Front Panel	A	
138	Power Supply, +5 VDC, +12 VDC, -12 VDC, +19VDC unreg., -19 VDC unreg.	L	
140	STD Computer Rack	N	
142	Cabinet	A	
150	System Power Switch	A	
152	Line Cord Assembly	K	
154	Transformer and Board Assembly (Rectifier)	B	LT-05R (part of)
156	Mounting Bracket	A	
158	Terminal Strip	E	
160	Fuse Holder	K	
162	Drive Motor (Wheel)	E	
164	Motor Plate	A	
166	Connector (Motor Power)	K	
168	Slide Switch (Motor)	K	
170	Drive Belt	F	
174	Wheel (Drive Pulley and Extension Ring)	A	
176	Shaft	A	
178, 184	Bearing (2)	M	
180, 186	Top and Bottom Bearing Plates	A	

192	Flange (Silver Soldered to Shaft)	A	
200-208	Slip Rings (5)	A	
210-218	Spring-Loaded Brush (5)	A	
220-228	Brush Holder Assy. (2)	A	
236	Connector and Lead Assembly	A	
238	Bracket for 240	A	
240	Wheel Position Circuit Board (Clock,Tach)	A	
242	Magnet	I	
244	Magnet Bracket	A	
246	Magnifier Mount Ring	A	
248	Radius Plate	A	
250	Prism Mount	A	
254	Prism	C	3309
256	Lens Mount	A	
258	Lens	C	
260	Photodiode or Phototransistor	V	
264	Carriage	A	
266	Track	A	
270	Guide Wheel (4)	J	7016-1
272	Spacer (4)	A	
274	Eccentric Bushing (4)	A	
282, 286	Track Block (2)	A	
284, 288	Riser Block (2)	A	
292	Guide Wheel (Ball Bearing)	M	
294	Eccentric Bushing	A	

298	Magnet	I	
302	Magnet (298) Mount	A	
306	Mount (2) for 308, 310	A	
308	Hall-effect Switch	H	276-1646
310	Hall-effect Switch Circuit Board	A	
312	Carriage Motor Bracket	A	
314,316	Carriage Motor	D	L92121-P2
318,379	Motor Control Board	A	
	Harness and Connector Ass'y		
320	Motor Control Circuit Board	A	
322	Skew Plate	A	
324	Shoulder Spacer (4)	A	
342	Plate (Glass, Mirror)	A	
344	Clip (2)	A	
350	Media Hold-down Spring	A	
356,364	Skew Motor	D	K92121-P2
366	Skew Lever	A	
370	Shoulder Spacer	A	
378	Skew Return Spring	A	
380	Radius Drive Motor	C	8813
382	Lead Screw (6-32)	A	
384	Lead Screw Pivot	A	
386	Radius Motor Control (FIG. 7)	A	
388	Laser Tube Assembly	B	
390	Laser Power Supply	B	LT-05R (part of)
392	Laser Cover	A	
394	21 mm Mirror	C	30,621
396	Mirror Mount	A	

400,402	Electro-optical Transducer Circuit	A
404	Fiber Optic Connector and Mount	G A
406	Fiber Optic	C
412,420	Fiber Optic Ferrule	G
413	Opto-Electric Transducer	V
414	Pulse Shaper Circuit	A
418	Electro-optical Transducer (LED)	V
424,426	4N25	V
Q1-Q10	1348	V
D1-D10	IN4001	V

TABLE II

SOURCE CODE	SOURCE
A	Special Design by Richard Ackerman for News Log International, Inc., P.O.Box 105, Fort Atkinson, WI 53538
B	CW Radiation, subsidiary, Aerotech, Inc., 101 Zeta Dr., Pittsburgh, PA 15238
C	American Science Center, 5700 Northwest Hgwy, Chicago, IL, 60646
D	Airpax, subsidiary of North American Phillips Co., Cheshire, CN. 06410
E	Lang & Epstein, Dayton St., Chicago, IL 606 22
F	Midwest Belting, 199 Gaylord Street, Elk Grove Village, IL 60007
G	AMP Inc., Pittsburgh, PA, or 1050 Morse, Elk Grove Village, IL 60007
H	Radio Shack, 9515 N. Milwaukee Ave. Niles, IL 60648
I	Micro Switch Division of Honeywell, Inc.
J	Stock Drive Products, 55 S. Denton Ave., New Hyde Park, NY 11040
K	Joseph Electronics Inc. 8830 N. Milwaukee Ave., Niles, IL 60648
L	Lab 1, News Log International, Inc., P.O.Box 105, Fort Atkinson, WI 53538
M	Harrison Supply Co., Milwaukee Ave., Wheeling, IL.
N	Pro Log Corp, 2411 Garden Ave., Monterey, CA 93940
V	Motorola, National, Semiconductor, Fairchild, Texas Instruments, et al

METHOD OF OPERATION

The system operates as follows, assuming all components have been placed in their operating conditions, wheel rotating, laser on, etc.

The optical system reads data to determine if the scan crosses data lines so that skew correction is required; skew is then corrected, by operating the skew motor and reading for a skew null.

The optical system reads into the computer which then operates the radius motor to lengthen the optical path until data is transmitted over the optical path indicating a datum from which incrementing may begin.

Subsequently, the data is scanned, row-by-row, and output to the computer for buffering and display. If an input addressing unit, e.g. keyboard, is employed, the supported data record is incremented until the correct data row is detected before data is output, decoded and displayed.

The following is the software for cyclic reading of a data record in which error correction may be employed using, for example, a Hamming code.

```
CP/M MACRO ASSEM 2.0     #001    READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
                         ;
                         ;
                         TITLE   'READ11 READER SOFTWARE 10/23/80
AAJ (C) 1980 NLI'
                         ;
                         MACLIB Z80
                         MACLIB Z80EXT
                         $*MACRO
                         PAGE 60
                         ;
                         ;
0000 =                   FALSE   EQU     0
FFFF =                   TRUE    EQU     NOT FALSE
                         ;
000D =                   CR      EQU     ODH
000A =                   LF      EQU     OAH
000C =                   FF      EQU     OCH
0009 =                   TAB     EQU     09H
0006 =                   ACK     EQU     06H
0015 =                   NAK     EQU     15H
0002 =                   STXT    EQU     02H
0004 =                   EOT     EQU     04H
0020 =                   SPACE   EQU     20H
                         ;
2100 =                   ORIGIN  EQU 2100H    ;LET'S EXECUTE FROM
                                               RAM.
2400 =                   INBLEN  EQU 2400H    ;INPUT TRACK BUFFER
                                               THIS LONG
2400 =                   PIXLEN  EQU 2400H    ;INPUT TRACK LENGTH
4000 =                   RAM     EQU 4000H
0003 =                   KBD     EQU 3
                         ;
0001 =                   PORT    EQU     1    ;LSB IS WHEEL SENSOR,
                                               MSB IS CARRIAGE
0000 =                   RDPORT  EQU     0    ;LSB IS PHOTO SENSOR
```

```
1   0000 =          STEPPER EQU    0      ;THIS IS STEPPER
                                           DRIVE PORT
    0002 =          FPORT   EQU    2      ;THIS IS FLAG
                                           OUTPUT PORT
5   015E #          FMARGIN SET    350    ;THIS IS MARGIN
                                           AT FRONT OF FILM
                                           ;(FRONT OF FILM
                                           IS SIDE WITH TITLE
                                           TRACK)
10  007D #          BMARGIN SET    125    ;THIS IS MARGIN
                                           AT BACK OF FILM
                    ;
                    ;STEPPER MOTOR CONTROLLER BITS ARE AS
                     FOLLOWS:
15                  ;
                    ;  D7  D6   D5   D4   D3   D2   D1   D0
                    ;
                    ;    SKEW MOTOR DRIVE   CARRIAGE MOTOR DRIVE
                    ;
20                  ;
    00D8 =          SKLIMIT EQU    216    ;THIS IS SKEW MOTOR
                                           TOTAL TRAVEL
    06F0 #          STEPS   SET    1776   ;THIS IS STEPS
                                           ACROSS ENTIRE FILM
25  0004 #          TRACK   SET    4      ;THIS MANY STEPS/
                                           TRACK
    01BC =          TRACKS  EQU    STEPS/TRACK ;THIS MANY
                                               TOTAL TRACKS
    0146 =          DATA    EQU TRACKS-(FMARGIN/TRACK)-
30                      (BMARGIN/TRACK)
                                           ;THIS MANY DATA
                                           TRACKS
                    ;
    FFFF =          HAMMING EQU    TRUE   ;SET TRUE TO ENABLE
35                                         ERROR CORRECTION
                    ;
```

```
1    0000 =          LEN60   EQU     FALSE ;THESE EQUATES
                                           SET LINE LENGTH
     FFFF =          LEN96   EQU     TRUE
     0000 =          LEN144  EQU     FALSE
5                    ;
                     IF LEN60
                     LLENGTH EQU 60
                     BITLEN EQU 17
```

```
1   CP/M MACRO ASSEM 2.0     #002   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                     ENDIF
                     ;
5                    IF LEN 96
    0060 =           LLENGTH EQU 96
    000A =           BITLENG EQU 10
                     ENDIF
                     ;
10                   IF LEN144
                     LLENGTH EQU 144
                     BITLEN  EQU 9
                     ENDIF
                     ;
15  0008 =           GROUPS  EQU    LLENGTH/12 ;HAMMING
                                                GROUPS
                     ;
                     ;
                     ;STEPPER MOTOR EQUATES
20                   ;
    FFFF =           TWOPHS  EQU    TRUE  ;THIS IS TWO-
                                           PHASE CLOCKING
                     ;
    0000 =           WAVE    EQU    FALSE ;THIS IS WAVE
25                                         CLOCKING
                     ;
    0000 =           HALFSTP EQU    FALSE ;THIS IS HALF-
                                           STEP CLOCKING
                     ;
30                   IF TWOPHS
                     ;
    000A =           STEP0   EQU    1010B
    0006 =           STEP1   EQU    0110B
    0005 =           STEP2   EQU    0101B
35  0009 =           STEP3   EQU    1001B
                     ;
                     ENDIF
```

```
                    IF WAVE
                    ;
                    STEP0   EQU     1000B
                    STEP1   EQU     0100B
                    STEP2   EQU     0010B
                    STEP3   EQU     0001B
                    ;
                    ENDIF
                    ;
                    IF HALFSTP
                    ;
                    STEP0   EQU     1000B
                    STEP1   EQU     1010B
                    STEP2   EQU     0010B
                    STEP3   EQU     0110B
                    STEP4   EQU     0100B
                    STEP5   EQU     0101B
                    STEP6   EQU     0001B
                    STEP7   EQU     1001B
                    ;
                    ENDIF
                    ;
0003 #              SDELAY  SET     03      ;THIS MANY MS DELAY
                                             FOR EACH STEP
                    ;
0087 #              TCONST  SET     135     ;THIS MANY LOOPS =
                                             1 MS AT 2.5 MHZ
                    ;
0195 #              STIMER  SET     SDELAY*TCONST
```

```
1   CP/M MACRO ASSEM 2.0     #003   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                     IF HALFSTP
                     ;
5                    FMARGIN SET FMARGIN*2
                     BMARGIN SET BMARGIN*2
                     TRACK   SET TRACK*2
                     STEPS   SET STEPS*2
                     STIMER  SET STIMER/2
10                   ;
                     ENDIF
                     ;
    FFFF =           ERRIND  EQU    TRUE   ;SET TRUE TO
                                            PRINT HAMMING
15                                          RESULTS
                     ;
    2100             ORG ORIGIN
                     ;
    2100 C37021      START:  JMP START1    ;AROUND COPYRIGHT
20                                          NOTICE
    2103 C37C2A              JMP SKEW     ;TO FORTRAN SKEW
                                            ROUTINE
    2106 C34025              JMP FSEEK    ;TO FORTRAN "SEEK"
    2109 C35625              JMP FGETCUR  ;TO FORTRAN "GET
25                                          CURRENT LINE"
    210C C3C228              JMP EJECT    ;TO FORTRAN EJECT
                                            ROUTINE
    210F C37A25              JMP FCLLINE  ;FORTRAN CLEAR LINE
                                            ROUTINE
30                                          ;LINE # IN C
    2112 C3692C              JMP CRT      ;FORTRAN PRINT
                                            ROUTINE FROM C
    2115 C37525              JMP FCLEAR   ;FORTRAN CLEAR
                                            SCREEN ROUTINE
35  2118 C35628              JMP KBDWAIT  ;FORTRAN KBD INPUT
                                            ROUTINE
```

```
    211B 434F505952          DB 'COPYRIGHT 1980 LAB1/NLI
                             POB 105 FORT ATKINSON,'
    2149 2057495343          DB ' WISCONSIN 53538.  ALL RIGHTS
                             RESERVED. '
2170 314040      START1: LXI SP,STACK ;SAFE STACK AREA
    2173 21182E              LXI H,SKTABLE ;POINT TO MIDDLE
                                           OF TABLE
    2176 22B240              SHLD SKADDR  ;STORE THE TABLE
                                           POINTER
2179 21A400                  LXI H,(8*16)+00100100B ;INIT
                                           THE BIT COUNTER
    217C 22B440              SHLD BITBYTE ;STORE THIS TOO
    217F 210000              LXI H,0      ;RESET THE TRACK
                                           COUNTER
2182 22B040                  SHLD CURLINE
2185 010100                  LXI B,1      ;SET SKEW POLARITY
                                           REGISTER
                             EXX
    2188+D9
2189 3EAA                    MVI A,STEP0+STEP0*16 ;SET THE
                                           STEPPER PORT UP
    218B D300                OUT STEPPER
                             LXIY 0       ;STORE STEP 0
    218D+FD21
218F+0000
    2191 210041              LXI H,INBUF  ;CLEAR BUFFERS
    2194 110141              LXI D,INBUF+1
    2197 010020              LXI B,2000H
    219A 3600                MVI M,0
                             LDIR
    219C+EDB0
    219E 3E0A                MVI A,BITLEN ;SET UP BIT
                                           LENGTH
    21A0 32A640              STA LEN
21A3 0E1A                    MVI C,CLR    ;SCREEN CLEAR
                                           CHARACTER
```

```
1    21A5 CD692C            CALL CRT      ;INIT THE CRT
                                          DRIVER
     21A8 DC692C            CALL CRT
                    ;
                    IF (ORIGIN EQ 0)
5                          JMP 0C00H     ;JUMP TO FORTRAN
                                          DIRECTLY IF
                                          ;THIS IS STAND-
                                          ALONE VERSION
                    ELSE
10                  ;
                    ;
                    ; CONSOLE COMMAND PROCESSOR
```

```
1   CP/M MACRO ASSEM 2.0     #004   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                        ;
                        ; THIS ROUTINE TAKES KEYBOARD INPUT AND
                          PERFORMS
5                       ; VARIOUS TASKS.  THE LEGAL COMMANDS
                          ARE:
                        ;
                        ;
                        ; [ , ] - STEP IN OR OUT 1 STEP
10                      ; < , > - SKEW IN OR OUT ONE STEP
                        ; H - HOME CARRIAGE TO FULLY IN POSITION
                        ; E - EJECT CARRIAGE TO FULLY OUT
                          POSITION
                        ; I - MOVE IN ONE TRACK
15                      ; O - MOVE OUT ONE TRACK
                        ; T - TYPE CURRENT TRACK
                        ; S - SKEW ADJUSTMENT ROUTINE
                        ; Q - QUIT TO ZAP
                        ; G (ADDR) - GOTO HEX TRACK ADDRESS
20                      ; N - RETRIEVE NEXT LINE
                        ; C - CENTER STEPPER DRIVES & INIT
                          POINTERS
                        ; Z - SKEW CHECK ROUTINE
                        ; W - WHAT TRACK ARE WE AT?
25                      ; X - PRINT UNCORRECTED LINE
                        ; - , + INCREMENT AND DECREMENT BIT SIZE
                        ; ? - QUERY PRESENT BIT SIZE
                        ; A - DISPLAY ALL TRACKS
                        ;
30                      ; OTHER CHARACTERS ARE IGNORED
                        ;
                        ;
    21AB 314040         WAIT:   LXI SP,STACK ;FIX STACK
    21AE CD4C28                 CALL KBDSTAT ;WATCH FOR CHARACTER
35                              JRZ WAIT     ;JUMP IF NONE
                                             WAITING
```

```
1   21B1+28F8
    21B3 CD5B28             CALL KBDIN    ;GET THE CHARACTER
    21B6 FE41               CPI 'A'       ;ALL TRACKS?
    21B8 C28A22             JNZ K90       ;JUMP IF NOT
    21BB CD7C2A             CALL SKEW     ;ADJUST TO TRACK 0
5   21BE C2C724             JNZ NOSKW     ;JUMP IF SKEW OUT
                                           OF RANGE
    21C1 CD2C29             CALL GETCUR   ;START W/ TRACK 0
    21C4 016001             LXI B,160H    ;THIS MANY TRACKS
    21C7 C5         K90LP:  PUSH B .      ;SAVE LINE COUNTER
10  21C8 114040             LXI D,OUTBUF  ;BUFFER IS HERE
    21CB AF                 XRA A         ;END THE STRING
                                           W/ NULL
    21CC 328040             STA OUTBUT+(GROUPS*8)
                    IF ERRIND
15  21CF 3AA940             LDA ERCOUNT   ;GET HAMMING SCORE
    21D2 B7                 ORA A         ;PERFECT?
                            JRZ NOER      ;JUMP IF PERFECT
                                           LINE
    21D3+2822
20  21D5 217E40             LXI H,OUTBUT+(GROUPS*8)-2;;END
                                           IT HERE
    21D8 3600               MVI M,0
    21DA CD5C2C             CALL CONPRNT
    21DD CD3325             CALL PLINE    ;PRINT THE HAMMING
25                                         MESSAGE
    21E0 2020455252         DB ' ERR = ',0
    21E9 3AA940             LDA ERCOUNT   ;GET THE ERROR
                                           COUNT
    21EC CD0025             CALL BASE10   ;CONVERT TO BASE 10
30  21EF CD1525             CALL PHEX     ;PRINT IT
    21F2 CD4228             CALL CRLF     ;PUT IN MISSING CR
                                           & LF
                            JMPR K77
    21F5+1803
35                  ENDIF
    21F7 CD5C2C     NOER:   CALL CONPRNT ;PRINT THE TRACK
```

```
1   CP/M MACRO ASSEM 2.0     #005   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    21FA DB03        K77:    IN KBD        ;GET KBD CHARACTER
    21FC E67F                ANI 07FH
    21FE FE03                CPA 'C'-40H   ;ABORT REQUEST?
5                            JRNZ K88      ;IGNORE OTHERS
    2200+2025
    2202 CD3325              CALL PLINE    ;TELL THEM...
    2205 0D0A2B2B2B          DB CR,LF,'++++ FUNCTION ABORTED ++++
                             ',CR,LF,0
10  2224 C3AB21              JMP WAIT
    2227 CD712A      K88:    CALL GETNXT   ;GET NEXT TRACK TO
                              BUFFER
                             JRZ K89       ;JUMP IF WE HAVE
                              GOOD STRING
15  222A+283A
    222C CD3325              CALL PLINE    ;OR ELSE FESS UP
    222F 0D0A504552          DB CR,LF,'PERMANENT ERROR ON
                             TRACK # ',0
    224D 2AB040              LHLD CURLINE ;THIS LINE #
20  2250 7C                  MOV A,H       ;PRINT HI BYTE
    2251 CD1525              CALL PHEX
    2254 7D                  MOV A,L       ;THEN LO BYTE
    2255 CD1525              CALL PHEX
    2258 CD4228              CALL CRLF
25  225B C1                  POP B         ;COUNT THE LINE
    225C 0B                  DCX B
    225D 78B1                MOV A,B ! ORA C ; FINISHED?
    225F CAAB21              JZ WAIT       ;KICK OUT IF SO
    2262 C5                  PUSH B        ;ELSE CONTINUE
30  2263 C32722              JMP K88
    2266 C1          K89:    POP B         ;CHECK THE LOOP
    2267 0B                  DCX B
    2268 78B1                MOV A,B ! ORA C
    226A C2C721              JNZ K90LP     ;LOOP TILL ALL
35                            TRACKS DISPLAYED
    226D CD3325              CALL PLINE
```

```
1   2270 414C4C2054          DB 'ALL TRACKS DISPLAYED',CR,
                             LF,0
    2287 C3AB21              JMP WAIT
    228A FE3F        K90:    CPI '?'        ;BIT SIZE QUERY?
                             JRZ PLNGTH     ;PRINT IT IF SO
5   228C+280B
    228E FE2D        K19:    CPA '-'        ;DECREMENT BIT SIZE?
                             JRNZ K20       ;JUMP IF NOT
    2290+2026
    2292 3AA640              LDA LEN        ;MAKE ONE SMALLER
10  2295 3D                  DCR A
    2296 32A640      LNGTH:  STA LEN
    2299 CD3325      PLNGTH: CALL PLINE
    229C 0D0A424954          DB CR,LF,'BIT LENGTH = ',0
    22AC 3AA640              LDA LEN
15  22AF CD1525              CALL PHEX
    22B2 CD4228              CALL CRLF
    22B5 C3AB21              JMP WAIT
    22B8 FE2B        K20:    CPI '+'        ;MAKE BIGGER?
                             JRNZ K21       ;JUMP IF NOT
20  22BA+2006
    22BC 3AA640              LDA LEN
    22BF 3C                  INR A
                             JMPR LNGTH     ;FINISH UP
    22C0+18D4
25  22C2 FE57        K21:    CPI 'W'        ;TRACK ADDRESS
                                             REQUEST?
    22C4 CA6423              JZ WTRK        ;JUMP IF SO
    22C7 FE58                CPI 'X'        ;PRINT UNCORRECTED
                                             LINE?
30                           JRNZ K12       ;JUMP IF NOT
    22C9+2061
    22CB CD8225              CALL INTRACK ;GET PIXELS
```

35

```
CP/M MACRO ASSEM 2.0     #006   READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
22CE CD8F26              CALL GETFTA  ;GET SYNC
                         JRNC PRNT    ;JUMP IF WE GOT IT

22D1+302C
22D3 CD3325              CALL PLINE   ;PRINT BAD NEWS
22D6 0D0A204341          DB CR,LF,' CAN''T GET BEGINNING
                         OF LINE SYNC',CR,LF,0

22FC C3AB21              JMP WAIT     ;TRY AGAIN
22FF CD7D29      PRNT:   CALL DECODE  ;DECODE TO BYTES
2302 214040              LXI H,OUTBUF ;THEY ARE HERE
2305 5D54                MOV E,L! MOV D,H ;COPY TO DE
2307 3E08                MVI A,8      ;8 BYTE GROUPS
2309 32A540              STA HBYTES
230C 3E08                MVI A,GROUPS ;HAMMING GROUPS TO A
230E CD9727              CALL HAMCODE ;DECODE STRING
2311 214040              LXI H,OUTBUF ;RESET POINTER
2314 0640                MVI B,GROUPS*8 ;THERE ARE THIS
                                          MANY
2316 7E          K13:    MOV A,M      ;CHECK THE BYTE
2317 E67F                ANI 7FH      ;DROP HI BIT
2319 FE20                CPA ' '      ;CONTROL CHARACTER?
                         JRNC K15     ;JUMP IF PRINTABLE

231B+3002
231D 3E20                MVI A,' '    ;IT IS GARBAGE
231F 4F          K15:    MOV C,A      ;GET THIS BYTE TO
                                       C REG

2320 23                  INX H        ;POINT TO NEXT
2321 CD692C              CALL CRT     ;TO CRT SCREEN
                         DJNZ K13     ;TILL DONE

2324+10F0
2326 CD4228              CALL CRLF    ;THEN CRLF
2329 C3AB21              JMP WAIT
232C FE5A        K12:    CPI 'Z'      ;SKEW TEST ?
232E C2B723              JNZ K11      ;JUMP IF NOT
```

```
    2331 CD512B            CALL SKEW2    ;DO FIRST SKEW
                                          ADJUST
                           JRNC ST1      ;JUMP IF SUCCESSFUL
    2334+3015
    2336 CD3325      ST4:  CALL PLINE    ;ELSE PRINT THE
                                          MESSAGE
    2339 0D0A204241        DB CR,LF,' BAD SKEW ', CR, LF,0
    2348 C3AB21            JMP WAIT      ;AND LOOP
    234B 0E2A        ST1:  MVI C,'*'     ;SHOW 1ST TRY
                                          WAS GOOD
    234D CD692C            CALL CRT
    2350 CD512B            CALL SKEW2    ;THEN TRY FOR 2ND
                                          ONE
                           JRC ST4       ;JUMP IF OUT OF
                                          RANGE
    2353+38E1
    2355 0E2A              MVI C,'*'     ;ELSE SHOW THAT
                                          THIS ONE GOOD
    2357 CD692C            CALL CRT
    235A CD512B            CALL SKEW2    ;DO FINAL ONE
                           JRC ST4       ;SHOW IF BAD
    235D+38D7
    235F 0E2A              MVI C,'*'     ;SHOW LAST ONE GOOD
    2361 CD692C            CALL CRT
    2364 CD8225      WTRK: CALL INTRAK   ;GET CURRENT TRACK
    2367 CD8F26      WTRK2:CALL GETFTA   ;WE GOT A FTA?
                           JRNC W2       ;JUMP IF SO
    236A+300D
    236C CD3325            CALL PLINE    ;TELL THEM WE AINT
                                          GOT
    236F 4E4F204654        DB 'NO FTA ',0
                           JMPR W3
    2377+180F
    2379 F5          W2:   PUSH PSW
    237A CD3325            CALL PLINE    ;PRINT MESSAGE
```

```
1    237D 465441203D  DB 'FTA = ',0
     2384 F1          POP PSW
```

```
CP/M MACRO ASSEM 2.0     #007   READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
2385 CD1525              CALL PHEX     ;PRINT THE FTA
2388 CD0F27      W3:     CALL GETBTA   ;TRY FOR BACK
                                        ADDRESS
                         JRNC W2       ;JUMP IF FOUND
238B+3011
238D CD3325              CALL PLINE    ;PRINT THE MESSAGE
2390 204E4F2042          DB ' NO BTA ',CR,LF,0
239B C3AB21              JMP WAIT
239E F5          W4:     PUSH PSW
239F CD3325              CALL PLINE
23A2 2042544120          DB ' BTA = ',0
23AA F1                  POP PSW
23AB CD1525              CALL PHEX     ;PRINT THE TRACK
23AE CD3325              CALL PLINE    ;END THE LINE
23B1 0D0A00              DB CR,LF,0
23B4 C3AB21              JMP WAIT
23B7 FE5B        K11:    CPA '['       ;STEP IN REQUEST?
                         JRNZ K1       ;JUMP IF NOT
23B9+2006
23BB CD7128              CALL INONE
23BE C3AB21              JMP WAIT      ;CONTINUE LOOP
23C1 FE5D        K1:     CPI ']'       ;STEP OUT REQUEST?
                         JRNZ K2       ;JUMP IF NOT
23C3+2003
23C5 CD6B28              CALL OUTONE
23C8 FE3C        K2:     CPI '<'       ;SKEW IN REQUEST?
                         JRNZ K3       ;JUMP IF NOT
23CA+2008
23CC 0601                MVI B,1       ;STEP IN 1
23CE CD6328              CALL SKEWIN
23D1 C3AB21              JMP WAIT      ;CONTINUE LOOP
23D4 FE3E        K3:     CPI '>'       ;SKEW OUT REQUEST?
                         JRNZ K14      ;JUMP IF NOT
```

```
23D6+2008
23D8 0601               MVI B,1       ;SKEW OUT ONE
23DA CD6728             CALL SKEWOUT
23DD C3AB21             JMP WAIT
23E0 FE43       K14:    CPI 'C'       ;CENTER & INIT?
                        JRNZ K4       ;JUMP IF NOT
23E2+2006
23E4 CDF228             CALL CENTER   ;DO IT
23E7 C3AB21             JMP WAIT      ;AND LOOP
23EA FE51       K4:     CPI 'Q'       ;QUIT REQUEST?
23EC CA0000             JZ 0000       ;REBOOT ON QUIT
                                       REQUEST
23EF FE49               CPI 'I'       ;IN TRACK REQUEST?
                        JRNZ K01      ;JUMP IF NOT
23F1+200D
23F3 2AB040             LHLD CURLINE  ;BUMP  THE LINE
                                       COUNT
23F6 23                 INX H
23F7 22B040             SHLD CURLINE
23FA CDA528             CALL INTRK
23FD C3AB21             JMP WAIT
2400 FE4F       K01:    CPI 'O'       ;OUT TRACK REQUEST?
                        JRNZ K02      ;JUMP IF NOT
2402+200D
2404 2AB040             LHLD CURLINE  ;GET LINE COUNT
2407 2B                 DCX H         ;ONE LESS
2408 22B040             SHLD CURLINE  ;
240B CDAD28             CALL OUTTRK
240E C3AB21             JMP WAIT
2411 FE45       K02:    CPI 'E'       ;EJECT REQUEST?
```

```
1   CP/M MACRO ASSEM 2.0      #008   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    2413 CCC228               CZ EJECT      ;CALL IF SO
    2416 FE48                 CPI 'H'       'HOME REQUEST?
    2418 CCDB28               CZ HOME       ;CALL IF SO
5   241B FE47                 CPI 'G'       ;GOTO LINE?
                              JRNZ K7       ;JUMP IF NOT
    241D+203C
    241F 4F                   MOV C,A       ;ECHO TO CRT
    2420 CD692C               CALL CRT
10  2423 210000               LXI H,0       ;CLEAR RESULT REGISTER
    2426 CD5628       GETADD: CALL KBDWAIT  ;GET CAHRACTER
    2429 4F                   MOV C,A       ;ECHO TO CRT
    242A CD692C               CALL CRT
    242D FE0D                 CPI CR        ;CARRIAGE RETURN?
15                            JRZ GOTADD    ;JUMP IF COMPLETE
    242F+2820
    2431 FE08                 CPI BS        ;BACK SPACE?
                              JRNZ K6       ;JUMP IF NOT
    2433+200C
20  2435 0604                 MVI B,4       ;BACK UP 4 BITS
                        K0:   RARR H        ;SHIFT HI BYTE
    2437+CB1C
                              RARR L        ;AND LO BYTE
    2439+CB1D
25                            RES 7,H       ;RESET HI BIT OF
                                             PAIR
    243B+CBBC
                              DJNZ K0       ;UNTIL DONE
    243D+10F8
30                            JMPR GETADD
    243F+18E5
    2441 D630           K6:   SUI '0'       ;REMOVE ASCII
                                             BIAS
    2443 FE0A                 CPI 10        ;TEST FOR DONE
35                            JRC ASCI      ;JUMP IF DONE
```

```
1   2445+3802
    2447 D607                SUI'A'-'9'-1 ;ELSE ADD CORRECTION
    2449 29292929    ASCI:   DAD H ! DAD H ! DAD H ! DAD H
                             ;SHIFT LEFT 4
    244D B5                  ORA L        ;PUT IN NYBBLE
5   244E 6F                  MOV L,A
                             JMPR GETADD  ;LOOP TILL CR FOUND
    244F+18D5
    2451 CD4228      GOTADD: CALL CRLF    ;FINISH LINE
    2454 EB                  XCHG         ;EXCHANGE TO PUT
10                           POINTER IN DE
    2455 CD372A              CALL GOLINE  ;GO THERE
    2458 C3AB21              JMP WAIT     ;AND LOOP
    2458 FE4E        K7:     CPI 'N'      ;GET NEXT LINE?
                             JRNZ K5      ;JUMP IF NOT
15  245D+2044
    245F CD712A              CALL GETNXT  ;GET NEXT LINE
                             JRZ K99      ;JUMP IF OK
    2462+2832
    2464 CD4228      K98:    CALL CRLF
20  2467 CD3325              CALL PLINE   ;TELL THEM.....
    246A 4355525245          DB 'CURRENT TRACK HAS PERMANENT
                             DATA ERROR',CR,LF,0
    2493 C3AB21              JMP WAIT
    2496 114040      K99:    LXI D,OUTBUF ;POINT OF BEGINNING
25  2499 AF                  XRA A        ;END THE STRING
    249A 328040              STA OUTBUF+(GROUPS*8)
    249D CD5C2C              CALL CONPRNT ;PRINT IT
    24A0 C3AB21              JMP WAIT
    24A3 FE54        K5:     CPI 'T'      ;PRINT THIS TRACK?
30                           JRNZ K8      ;JUMP IF NO
```

```
CP/M MACRO ASSEM 2.0      #009   READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
24A5+2015
24A7 CD2C29               CALL GETCUR  ;GET THE LINE
                          JRNZ K98
24AA+20B8
24AC 114040               LXI D,OUTBUF ;POINT TO BEGINNING
24AF AF                   XRA A        ;END THE STRING
24B0 328040               STA OUTBUF+(GROUPS*8)
24B3 CD5C2C               CALL CONPRNT ;PRINT IT
24B6 CD4228               CALL CRLF    ;ADD CARRIAGE
                                        RETURN & LINE FEED
24B9 C3AB21               JMP WAIT     ;RUN LOOP
24BC FE53        K8:      CPI 'S'      ;SKEW FUNCTION
                                        REQUEST?
24BE C2AB21               JNZ WAIT     ;JUMP IF NOT
24C1 CD7C2A               CALL SKEW    ;DO SKEW ROUTINE
24C4 CA6723               JZ WTRK2     ;SHOW ADDRESSES IF
                                        SUCCESSFUL
24C7 CD3325      NOSKW:   CALL PLINE   ;ELSE PRINT THE
                                        ERROR
24CA 0D0A534B45           DB CR,LF,'SKEW NOT WITHIN RANGE.
                          PLEASE REPOSITION FILM.',CR,LF,
24FD C3AB21               JMP WAIT
                ;
2500 0600        BASE10:  MVI B,0      ;THIS IS 10S
                                        COUNTER
2502 FE0A        B10:     CPI 10       ;TENS GROUP?
                          JRC ADONE    ;JUMP IF NONE LEFT
2504+3805
2506 D60A                 SUI 10       ;ELSE SUBTRACT IT
2508 04                   INR B        ;AND SAVE IT
                          JMPR B10
2509+18F7
250B 4F          ADONE:   MOV C,A      ;SAVE THE ONES
250C 78                   MOV A,B      ;GET THE TENS
250D 17171717             RAL ! RAL !RAL ! RAL ;TO HI NYBBLE
2511 E6F0                 ANI 0F0H
2513 81                   ADD C
```

```
2514 C9                 RET
                        ;
                        ;
2515 47          PHEX:  MOV B,A       ;SAVE BYTE IN B
2516 1F1F1F1F           RAR ! RAR ! RAR ! RAR ;GET TOP
                                              HALF
251A CD2925             CALL HEXASC  ;TO ASCII
251D 4F                 MOV C,A      ;PRINT IT
251E CD692C             CALL CRT
2521 78                 MOV A,B      ;THEN DO LO HALF
2522 CD2925             CALL HEXASC
2525 4F                 MOV C,A
2526 C3692C             JMP CRT
                        ;
2529 E60F        HEXASC: ANI 0FH      ;DROP HI BITS
252B C630               ADI '0'      ;ADD ASCII OFFSET
252D FE3A               CPA '9'+1    ;IS IT A LETTER?
252F D8                 RC           ;DONE IF NO
2530 C607               ADI 7        ;ELSE ADD REMAINDER
2532 C9                 RET
                        ;
                        ENDIF
                        ;
2533 E1          PLINE: POP H        ;GET MESSAGE ADDRESS
2534 7E          PL2:   MOV A,M      ;GET NEXT CHARACTER
2535 23                 INX H        ;POINT TO NEXT
2536 B7                 ORA A        ;DONE?
                        JRNZ PL1     ;JUMP IF NOT
2537+2001
2539 E9                 PCHL         ;RETURN IF SO
```

```
CP/M MACRO ASSEM 2.0     #010   READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
253A 4F          PL1:    MOV C,A       ;PASS CHARACTER IN C
253B CD692C              CALL CRT      ;TO THE SCREEN
                         JMPR PL2      ;LOOP TILL ALL
                                        PRINTED
253E+18F4
                 ;
                 ;
                 ; SUBROUTINE FSEEK
                 ;
                 ; SUBROUTINE FSEEK IS THE FORTRAN
                   INTERFACE FOR
                 ; SEEKING, READING & DECODING A TRACK
                   FROM FILM.
                 ; THE DESIRED TRACK IS POINTED TO BY
                   THE (DE) PAIR
                 ; AND AN ERROR INDICATION IS RETURNED
                   IN THE A REGISTER.
                 ; THE RETRIEVED TRACK IS STORED AT
                   OUTBUF.
                 ;
2540 CD372A      FSEEK:  CALL GOLINE   ;TRY TO GO THERE
2543 B7                  ORA A
                         JRZ FSEEK1    ;JUMP IF GOOD
2544+2806
2546 3EFF        SEEKERR:MVI A,OFFH    ;GET BAD RETRIEVE
                                        INDICATION
2548 32A840              STA ERRFLAG   ;SHOW IT
254B C9                  RET
254C CD2C29      FSEEK1: CALL GETCUR   ;GET THE LINE
254F B7                  ORA A
                         JRNZ SEEKERR ;JUMP IF BAD
                                        RETRIEVE
2550+20F4
2552 32A840              STA ERRFLAG   ;SHOW GOOD RETRIEVE
```

```
1   2555 C9                 RET
                    ;
                    ; SUBROUTINE FGETSUR
                    ;
                    ; SUBROUTINE FGETCUR IS THE FORTRAN
5                     INTERFACE FOR
                    ; DOING FORMATTED READS OF THE CURRENTLY
                      READ TRACK.
                    ; IT POINTS TO THE RETRIEVED DATA WITH
                      THE (DE) PAIR
10                  ; AND RETURNS AN ERROR INDICATION IN
                      THE A REGISTER.
                    ;
    2556 114040     FGETSUR:LXI D,OUTBUF ;RECOVERED DATA IS
                                          HERE
15  2559 3AA840             LDA ERRFLAG  ;ERROR INDICATION
                                          IS HERE
    255C B7                 ORA A        ;SET THE FLAGS
                            JRZ GETOK    ;JUMP IF OK
    255D+2802
20  255F AF                 XRA A        ;SHOW NO BUFFER
    2560 C9                 RET
    2561 3E40       GET OK: MVI A,GROUPS*8 ;LINE LENGTH TO
                                             A
    2563 C9                 RET
25                  ;
                    ;
                    ; SUBROUTINE TOLINE
                    ;
                    ; SUBROUTINE  TOLINE SETS UP THE CRT
30                    CURSOR TO THE
                    ; LINE POINTED TO BY THE C REGISTER
                    ;
    2564 3E02       TOLINE: MVI A,2      ;GET THE CURSOR
                                          ADDRESS MODE
35
```

```
1   2566 3200E4          STA CFLAG
    2569 79              MOV A,C       ;GET THE LINE NUMBER
    256A C61F            ADI SPACE-1   ;ADD NECESSARY
                                        OFFSET
    256C 4F              MOV C,A       ;PASS IN C REGISTER
5   256D CD692C          CALL CRT      ;TO THE CRT DRIVER
    2570 0E20            MVI C,SPACE   ;GET COLUMN 0
    2572 C3692C          JMP CRT       ;DO IT & RETURN
```

```
1   CP/M MACRO ASSEM 2.0     #011   READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                         ;
                         ; SUBROUTINE FCLEAR
                         ;
5                        ; SUBROUTINE FCLEAR CLEARS AND INITS
                           THE CRT SCREEN
                         ;
    2575 0E1A            FCLEAR: MVI C,CLR    ;THIS SIMPLE
    2577 C3692C                  JMP CRT
10                       ;
                         ;
                         ; SUBROUTINE FCLLINE
                         ;
                         ; SUBROUTINE FCLLINE CLEARS THE CRT LINE
15                         POINTED TO
                         ; BY THE C REGISTER
                         ;
    257A CD6425          FCLLINE:CALL TOLINE  ;GO THERE
    257D 0E15                    MVI C,NAK    ;CLEAR THE LINE
20  257F C3692C                  JMP CRT      ;AND RETURN
                         ;
                         ;
                         ;
                         ;INTRAK WILL INPUT A WHOLE TRACK WAVEFORM
25                        OF ONES AND ZEROES.
                         ;
    2582 E5              INTRAK: PUSH H       ;SAVE USER REGISTERS
    2583 C5                      PUSH B
    2584 DB01            ITK:    IN PORT      ;GET THE READ PORT
30  2586 1F                      RAR          ;MAG. SENSOR BIT
                                               TO CY
                                 JRC ITK      ;WAIT FOR ACTIVE
                                               SENSOR
    2587+38FB
35
```

```
                    ;
                    ; WE WILL NOW DELAY UNTIL THE READ HEAD
                      REACHES THE
                    ; BEGINNING OF THE TRACK.
                    ;
2589 219501               LXI H,TCONST*3;ABOUT 5MS TO
                                        TRACK START
258C 2B           DELAY:  DCX H         ;COUNT THE STEP
258D 7C                   MOV A,H       ;CHECK FOR DONE
258E B5                   ORA L
                          JRNZ DELAY    ;LOOP TILL TIMEOUT
258F+20FB
                    ;
                    ; WE WILL NOW MAP SAMPLES INTO RAM
                      BUFFER.  THE BUFFER
                    ; IS QUITE LONG, AS THE INPUT ROUTINE
                      MAPS BETWEEN 5
                    ; AND 11 SAMPLES PER TRANSITION FOR LINE
                      LENGTH OF 96
                    ;
                    ;READ IN A 12KB BUFFER FROM SCANNER.
                    ;
2591 D302                 OUT FPORT     ;SEND PULSE TO FLAG
                                        PORT
2593 2F                   CMA
2594 D302                 OUT FPORT
2596 2F                   CMA
2597 D302                 OUT FPORT
2599 210041               LXI    H,INBUF ;SETUP DESTINATION
                                         POINTER
259C 0E00                 MVI    C,RDPORT ;SETUP I/O
                                          SOURCE POINTER.
259E 0600         INLOOP: MVI    B,0    ;CLEAR TRANSFER
                                        COUNTER FOR 256
                                        BYTES.
```

```
1                       REPT    39      ;THIS SHOULD BE A
                                        12K BUFFER.
                        INIR
                        NOP
                        ENDM

5       25A0+EDB2
        25A2+00
        25A3+EDB2
```

```
1   CP/M MACRO ASSEM 2.0     #012  READ 11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    25A5+00
    25A6+EDB2
    25A8+00
5   25A9+EDB2
    25AB+00
    25AC+EDB2
    25AE+00
    25AF+EDB2
10  25B1+00
    25B2+EDB2
    25B4+00
    25B5+EDB2
    25B7+00
15  25B8+EDB2
    25BA+00
    25BB+EDB2
    25BD+00
    25BE+EDB2
20  25C0+00
    25C1+EDB2
    25C3+00
    25C4+EDB2
    25C6+00
25  25C7+EDB2
    25C9+00
    25CA+EDB2
    25CC+00
    25CD+EDB2
30  25CF+00
    25D0+EDB2
    25D2+00
    25D3+EDB2
```

```
25D5+00
25D6+EDB2
25D8+00
25D9+EDB2
25DB+00
25DB+EDB2
25DE+00
25DF+EDB2
25E1+00
25E2+EDB2
25E4+00
25E5+EDB2
25E7+00
25E8+EDB2
25EA+00
25EB+EDB2
25ED+00
25EE+EDB2
25F0+00
25F1+EDB2
25F3+00
25F4+EDB2
25F6+00
25F7+EDB2
25F9+00
25FA+EDB2
25FC+00
25FD+EDB2
```

```
1   CP/M MACRO ASSEM 2.0     #013  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    25FF+00
    2600+EDB2
    2602+00
5   2603+EDB2
    2605+00
    2606+EDB2
    2608+00
    2609+EDB2
10  260B+00
    260C+EDB2
    260E+00
    260F+EDB2
    2611+00
15  2612+EDB2
    2614+00
    2615 AF                  XRA A          ;SEND END PULSE TO
                                             FLAG PORT
    2616 D302                OUT FPORT
20  2618 2F                  CMA
    2619 D302                OUT FPORT
    261B 2F                  CMA
    261C D302                OUT FPORT
    261E C1                  POP B
25  261F E1                  POP H
    2620 C9                  RET
                        ;
                        ;
                        ; SUBROUTINE GETLAST
30                      ;        ;THIS SUBROUTINE IS IDENTICAL
                                  TO (AND IN FACT JUMPS
                        ;        ;TO) GETBIT WITH THE EXCEPTION
                                  THAT IT DECODES BITS
                        ;        ;FROM THE BACK OF THE PIXEL
35                                BUFFER
                        ;        ;TOWARDS THE FRONT OF THE
                                  BUFFER.
                        ;
```

```
2621 C5          GETLAST:PUSH B
2622 D5                  PUSH D
2623 4E                  MOV C,M
2624 AF                  XRA A          ;CLEAR CY.
2625 57                  MOV D,A        ;CLEAR HI HALF OF
                                         WORD
2626 3AA640              LDA LEN        ;GET BIT LENGTH
2629 5F                  MOV E,A        ;TO COMPLETE THE
                                         WORD
                         DSBC D         ;SUBTRACT A BITLENGTH
                                         DISTANCE FROM HL.
262A+ED52
262C 22AE40              SHLD SAVHL     ;EXPECTED POSITION
                                         OF NEXT TRANSISTION.
                         JMPR GET3      ;GOT TO GETBIT CODE
                                         AND CONTINUE.
262F+180D
                 ;
                 ;
                 ;
                 ; SUBROUTINE GETBIT
                 ;
                 ;THIS ROUTINE IS A MORE SOPHISTICATED
                  DECODING PROGRAM
                 ;TO ALLOW THE READING OF ASYMETRICAL
                  DOT PATTERNS.
                 ;
2631 C5          GETBIT: PUSH B         ;SAVE B&C FOR USER
2632 D5                  PUSH D         ;ALSO D&E
2633 4E                  MOV C,M        ;SAVE CURRENT POLARITY
2634 1600                MVI D,0        ;CLEAR HI HALF OF
                                         WORD
2636 3AA640              LDA LEN        ;PLUS LENGTH COUNTER
2639 5F                  MOV E,A        ;COMPLETES THE WORD
```

```
1   263A 19                 DAD D
    263B 22AE40             SHLD SAVHL    ;EXPECTED NEXT BIT
                                           POSITION
```

```
1   CP/M MACRO ASSEM 2.0      #014  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    263E 110100       GET3:   LXI D,1       ;SEARCH COUNTER IN
                                             D
    2641 2B                   DCX H         ;GET LAST CELL
5   2642 7E                   MOV A,M
    2643 23                   INX H         ;POINT TO THIS
                                             CELL
    2644 BE                   CMP M         ;SAME?
                              JRNZ TFOUND   ;JUMP IF SO
10  2645+202A
    2647 3AA640               LDA LEN       ;GET THE BYTE
    264A 1F                   RAR           ;DIVIDE BY 2
    264B E67F                 ANI 07FH      ;DROP HI BIT
    264D 47                   MOV B,A       ;THIS IS SEARCH
                                             DISTANCE
15
    264E AF           TRLOOP: XRA A         ;CLEAR CARRY
                              DSBC D        ;SUBTRACT SEARCH
                                             LENGTH
    264F+ED52
20  2651 2B                   DCX H         ;GET LAST CELL
    2652 7E                   MOV A,M
    2653 23                   INX H         ;POINT TO THIS
                                             CELL
    2654 BE                   CMP M         ;SAME?
25                            JRNZ TFOUND   ;JUMP IF SO
    2655+201A
    2657 19                   DAD D         ;TRY AHEAD SAME
                                             AMOUNT
    2658 19                   DAD D
30  2659 2B                   DCX H         ;GET LAST CELL
    265A 7E                   MOV A,M
    265B 23                   INX H         ;POINT TO THIS
                                             CELL
    265C BE                   CMP M         ;SAME?
35                            JRNZ TFOUND   ;JUMP IF SO
    265D+2012
    265F AF                   XRA A         ;ELSE CENTER IT UP
                              DSBC D
```

```
1   2660+ED52
    2622 1C                 INR E          ;BUMP SEARCH LENGTH
                            DJNZ TRLOOP    ;DO TILL TRANSISTION
                                            FOUND
    2663+10E9
5   2665 2AAE40             LHLD SAVHL     ;PRETEND THAT A
                                            LONG TRANSISTION
                                            WAS FOUND
    2668 AF                 XRA A          ;IT HAS TO BE A
                                            ZERO, SO CY=0
10                                          AT RETURN.
    2669 3D                 DCR A          ;WE SET NON-ZERO
                                            FLAG TO INDICATE
                                            GARBAGE
    266A 3E00               MVI A,o
15  266C 37                 STC            ;RESET CY
    266D 3F                 CMC
    266E D1                 POP D
    266F C1                 POP B
    2670 C9                 RET
20  2671 3E65       TFOUND: MVI A,((INBUF SHR 8)+(INBLEN
                            SHR 8))        ;CHECK FOR TOP OF
                                            BUFF
                                           ;A SHOULD = 42H +
                                            30H = 72H
25  2673 BC                 CMP H          ;ARE WE THERE YET?
                            JRNZ T2        ;JUMP IF NOT
    2674+200D
    2676 AF                 XRA   A        ;CLEAR CARRY.
    2677 57                 MOV D,A        ;FORM THE BIT
30                                          LENGTH WORD
    2678 3AA640             LDA LEN        ;FROM STORED BYTE
    267B 5F                 MOV E,A
    267C AF                 XRA A
                            DSBC D         ;BACKUP HL BY ONE
35                                          BITLENGTH.
    267D+ED52
```

```
                267F D1             POP D       ;RESTORE REGISTERS
                2680 C1             POP B
                2681 3D             DCR   A     ;01 A=FF FOR BUF
                                                 OVERFLOW INDICATOR.
                2682 C9             RET
                2683 7E       T2:   MOV A,M     ;GET CURRENT
                                                 POLARITY
```

```
CP/M MACRO ASSEM 2.0     #015  READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
2684 B9                  CMP C          ;SAME AS OLD
                                         POLARITY?
                         JRNZ ZERO      ;JUMP IF ZERO
2685+2004
2687 37                  STC            ;RETURN WITH CY
                                         SET
2688 D1                  POP D
2689 C1                  POP B
268A C9                  RET
268D AF          ZERO:   XRA A          ;0 TO CY
268C D1                  POP D
268D C1                  POP B
268E C9                  RET
                 ;
                 ;
                 ; SUBROUTINE GETFTA
                 ;
                 ; SUBROUTINE GETFTA RECOVERS THE FRONT
                   TRACK ADDRESS
                 ; FROM INBUF.  IF SUCCESSFUL, IT RETURNS
                   WITH THE
                 ; ADDRESS IN THE A REGISTER, AND THE
                   CARRY FLAG RESET.
                 ; IF UNSUCCESSFUL, IT RETURNS WITH THE
                   CARRY FLAG
                 ; SET AND GARBAGE IN THE A REGISTER.
                 ;
                 ; UPON SUCCESSFUL COMPLETION OF A ADDRESS
                   FIND, IT WILL
                 ; ALSO STORE THE BEGINNING OF DATA AREA
                   POINTER AT LSTART.
                 ;
```

```
                        ; THIS ROUTINE STOMPS THE AF PAIR.
                        ;
268F 3E0A       GETFTA: MVI A, BITLEN;GET NORMAL LENGTH
2691 32A640             STA LEN
2694 C5         FTA:    PUSH B          ;SAVE USER B&C
2695 D5                 PUSH D
2696 E5                 PUSH H          ;AND H&L
2697 0E40               MVI C,64        ;LIMIT TO C REGISTER
2699 210041             LXI H,INBUF     ;PIXELS ARE HERE
269C 0608       LDR:    MVI B,8         ;FIRST WE GET AT
                                         LEAST
                                        ;8 ZEROS IN A ROW
269E 0D         LDR2:   DCR C           ;COUNT LEADER LENGTH
                        HRZ BADFTA      ;KICK OUT IF TOO FAR
269F+2864
26A1 CD3126             CALL GETBIT     ;GET LEADER BIT
                        JRC LDR         ;RESET COUNTER IF
                                         1 FOUND
26A4+38F6
                        JRNZ LDR        ;OR IF GARBAGE BIT
                                         FOUND
26A6+20F4
                        DJNZ LDR2       ;LOOP FOR 8 ZEROS
26A8+10F4
26AA 48                 MOV C,B         ;RESET COUNTER
26AB 0D         LDR3:   DCR D           ;COUNT BITS
                        JRZ BADFTA      ;KICK OUT IF TOO
                                         LONG
26AC+2857
26AE CD3126             CALL GETBIT     ;LOOP TILL ONE FOUND
                        JRNC LDR3       ;
26B1+30F8
26B3 22AA40             SHLD POINTER    ;SAVE POINTER IN
                                         CASE OF GARBAGE
2686 22AA40             PUSH H          ;ALSO IN STACK
26B7 214040             LXI H,OUTBUF    ;GET THE FIRST 3
                                         BYTES
```

```
1    26BA E3                    XTHL          ;TO OUTBUF
     26BB 1E03                  MVI E,3       ;
     26BD 0608          FTA1:   MVI B,8       ;8 BITS/BYTE
     26BF CD3126        FTA2:   CALL GETBIT   ;GET FTA TO D
                                RALR D        ;ONE BIT AT A TIME
```

```
1   CP/M MACRO ASSEM 2.0      #016  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    26C2+CB12
                              DJNZ FTA2     ;TILL DONE
    26C4+10F9
5   26C6 E3                   XTHL          ;OUTPUT POINTER
                                             BACK
    26C7 72                   MOV M,D       ;PUT CHARACTER
                                             AWAY
    26C8 23                   INX H         ;NEXT BUFFER POSITION
10  26C9 E3                   XTHL          ;INPUT POINTER BACK
    26CA 1D                   DCR E         ;COUNT BYTE NO.
                              JRNZ FTA1     ;LOOP TILL ALL
                                             BYTES FOUND
    26CB+20F0
15  27CD E3                   XTHL          ;GET THE INPUT
                                             POINTER BACK
    26CE E1                   POP H         ;RECOVER THE PIXEL
                                             POINTER
    26CF CD3126               CALL GETBIT   ;THIS MUST BE A ONE
20                            JRNC FTA4     ;JUMP IF NOT
    26D2+3036
    26D4 0608                 MVI B,8       ;PLUS 8 ZEROS
    26D6 CD3126       FTA3:   CALL GETBIT   ;WE WATCH FOR 8
                                             ZEROS
25                            JRC FTA4      ;JUMP IF A ONE
                                             FOUND
    26D9+382F
                              DJNZ FTA3     ;LOOP FOR EIGHT
                                             ZEROS
30  26DB+10F9
    26DD 22AC40               SHLD LSTART   ;STORE START OF
                                             DATA LINE
    26E0 114040       ADDONE: LXI D,OUTBUF  ;THIS IS HAMMING
                                             BUFFER
35
```

```
1   26E3 626B              MOV H,D ! MOV L,E ;IN AND OUT
                                               BUFFERS ARE
                                               THE SAME
    26E5 3E02              MVI A,2           ;2 BYTE BLOCK TO
                                               DECODE
5   26E7 32A540            STA HBYTES
    26EA 3E01              MVI A,1           ;ONE SUCH BLOCK
    26EC CD9727            CALL HAMCODE      ;DECODE IT
    26EF 3A4140            LDA OUTBUT+1      ;GET THE BACKWARDS
                                               BYTE
10  26F2 0608              MVI B,8           ;TURN IT RIGHT WAY
                                               AROUND
    26F4 1F        FTA5:   RAR               ;SHIFT ONE BIT AT
                                               A TIME..
                           RALR D            ;THROUGH CY TO D
15                                             REGISTER
    26F5+CB12
                           DJNZ FTA5
    26F7+10FB
    26F9 3A4040            LDA OUTBUF        ;GET FIRST BYTE
20  26FC BA                CMP D             ;CHECK FOR CORRECT
                           JRNZ FTA4         ;JUMP IF GARBAGE
    26FD+200B
    26FF AF                XRA A             ;CLEAR CARRY FLAG
    2700 7A                MOV A,D           ;PASS ADDRESS IN A
25                                             REGISTER
    2701 E1                POP H             ;RESTORE USER REGS
    2702 D1                POP D
    2703 C1                POP B
    2704 C9                RET
30  2705 E1        BADFTA: POP H
    2706 D1                POP D
```

```
    2702 C1                 POP B          ;RESTORE USDER REGS
    2708 37                 STC            ;SHOW BAD FTA
    2709 C9                 RET
    270A 2AAA40     FTA4:   LHLD POINTER   ;RESET OUR MEMORY
                                            POINTER
                            JMPR LDR       ;AND TRY AGAIN
    270D+188D
                    ;
                    ;
                    ;
                    ; SUBROUTINE GETBTA
                    ;
                    ; SUBROUTINE GETBTA RECOVERS THE BACK
                      TRACK ADDRESS
                    ; FROM INBUF.  IF SUCCESSFUL, IT RETURNS
                      WITH THE
```

```
1   CP/M MACRO ASSEM 2.0     #017  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                     ; ADDRESS IN THE A REGISTER AND THE CARRY
                       FLAG RESET.
                     ; IF UNSUCCESSFUL, IT RETURNS WITH THE
5                      CARRY FLAG
                     ; SET AND GARBAGE IN THE A REGISTER.
                     ;
                     ; THIS ROUTINE REQUIRES THE EQUATE
                       "PIXLEN" WHICH IS
10                   ; THE NUMBER OF BYTES FILLED WITH
                       PIXELS FROM THE
                     ; START OF FILM TO THE END OF FILM.
                       THIS IS INDEPENDANT
                     ; OF DATA DENSITY, BUT CHANGES WITH
15                     CPU CLOCK FREQ.
                     ; AND ALSO WITH WHEEL ROTATION SPEED.
                     ;
                     ; THIS ROUTINE STOMPS THE HL PAIR AND
                       THE AF PAIR.
20                   ;
    270F 3E0A        GETBTA: MVI A,BITLEN ;TRY ONE LESS
    2711 3D                  DCR A
    2712 32A640              STA LEN
    2715 CD2027              CALL BTA
25  2718 D0                  RNC
    2719 3AA640              LDA LEN      ;OR ONE MORE
    271C 3C                  INR A
    271D 32A640
                     ;
30  2720 C5          BTA:    PUSH B       ;SAVE USER B&C
    2721 D5                  PUSH D
    2722 E5                  PUSH H       ;ALSO H&L
    2723 0E40                MVI C,64     ;LIMIT TO C REGISTER
                                           AS LIMIT
35  2725 210065              LXI H,INBUF+PIXLEN ;END OF LINE
                                                IS HERE
    2728 0608        BTA1:   MVI B,8      ;AT LEAST 8 ZEROS
                                           TO START
```

```
    272A 0D          BTA2:   DCR C          ;COUNT BITS
                             JRZ BADBTA     ;JUMP IF TOO FAR
    272B+2860
    272D CD2126              CALL GETLAST   ;GET NEXT BIT
                             JRC BTA1       ;LOOP & RESET COUNT
                                             IF 1 FOUND
    2730+38F6
                             JRNZ BTA1      ;ALSO IF GARBAGE
                                             BIT FOUND
    2732+20F4
                             DJNZ BTA2      ;ELSE LOOP FOR 8
                                             ZEROS
    2734+10F4
    2736 0D          BTA3:   DCR C          ;COUNT BITS
                             JRZ BADBTA     ;KICK OUT IF TOO
                                             MANY
    2737+2854
    2739 CD2126              CALL GETLAST   ;CHECK FOR FLAG
                             JRNC BTA3      ;LOOP TILL FOUND
    273C+30F8
    273E 22AA40              SHLD POINTER   ;SAVE POINTER IN
                                             CASE WE GOT GARBAGE
    2741 E5                  PUSH H         ;ALSO IN STACK
    2742 214040              LXI H,OUTBUF   ;THIS IS HAMMING
                                             BUFFER
    2745 E3                  XTHL           ;GET INPUT POINTER
    2746 1E03                MVI E,3        ;DO 3 BYTES
    2748 0608        BTA0:   MVI B,8        ;8 BITS/BYTE
    274A CD2126      BTA4:   CALL GETLAST   ;GET TRACK ADDRESS
                             RALR D         ;TO D REGISTER
    274D+CB12
                             DJZN BTA4      ;ONE BIT AT A TIME
    274F+10F9
    2751 E3                  XTHL           ;OUTPUT POINTER
    2752 72                  MOV M,D        ;SAVE THE RESULTANT
                                             BYTE
```

```
1    2753 23          INX H        ;NEXT BUFFER POSITION
     2754 E3          XTHL         ;GET INPUT POINTER
     2755 1D          DCR E        ;COUNT BYTE NUMBER
                      JRNZ BTA0    ;LOOP FOR 3 BYTES
```

```
1   CP/M MACRO ASSEM 2.0     #018  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    2756+20F0
    2758 E3                  XTHL          ;PIXEL POINTER TO
                                            STACK
5   2759 E1                  POP H         ;RECOVER PIXEL
                                            POINTER
    275A CD2126              CALL GETLAST  ;END FLAG THERE?
                             JRNC BTA6     ;JUMP IF END FLAG
                                            MISSING
10  275D+3033
    275F 0608                MVI B,8       ;WE NEED 8 MORE
                                            ZEROS
    2761 CD2126      BTA5:   CALL GETLAST  ;GET NEXT BIT
                             JRC BTA6      ;JUMP IF A ONE IS
15                                          FOUND
    2764+382C
                             DJNZ BTA5     ;LOOP TILL 8 ZEROS
                                            ARE FOUND
    2766+10F9
20  2768 114040              LXI D,OUTBUF  ;THIS IS HAMMING
                                            BUFFER
    276B 626B                MOV H,D ! MOV L,E ;IN AND OUT
                                                BUFFERS ARE
                                                THE SAME
25  376D 3E02                MVI A,2       ;2 BYTE BLOCK TO
                                            DECODE
    276F 32A540              STA HBYTES
    2772 3E01                MVI A,1       ;ONE SUCH BLOCK
    2774 CD9727              CALL HAMCODE  ;DECODE IT
30  2777 3A4140              LDA OUTBUF+1  ;GET THE BACKWARDS
                                            BYTE
    277A 0608                MVI B,8       ;TURN IT RIGHT WAY
                                            AROUND

35
```

```
277C 1F          BTA9:   RAR             ;SHIFT ONE BIT AT
                                          A TIME..
                         RALR D          ;THROUGH CY TO D
                                          REGISTER
277D+CB12
                         DJNZ BTA9
277F+10FB
2781 3A4040              LDA OUTBUF      ;GET FIRST BYTE
2784 BA                  CMP D           ;CHECK FOR CORRECT
                         JRNZ BTA6       ;JUMP IF GARBAGE
2785+200B
2787 AF                  XRA A           ;CLEAR CARRY FLAG
2788 7A                  MOV A,D         ;PASS ADDRESS IN
                                          A REGISTER
2789 E1                  POP H           ;RESTORE USER REGS
278A D1                  POP D
278B C1                  POP 8
278C C9                  RET
                 ;
278D E1          BADBTA: POP H
278E D1                  POP D
278F C1                  POP B           ;RESTORE USER B&C
2790 37                  STC             ;SHOW BAD BTA
2791 C9                  RET
2792 2AAA40      BTA6:   LHLD POINTER    ;GET THE OLD BUFFER
                                          POINTER
                         JMPR BTA1       ;AND LOOP FOR NEW
                                          TRY
2795+1891
                 ;
                 ;
                 ; HAMMING ERROR CORRECTION DECODER
                   MODULE
                 ;
                 ; THIS MODULE DECODES AND PERFORMS
                   ERROR CORRECTION
```

```
1                ; ON A (12,8,3) HAMMING CODE INTERLEAVED
                   TO DEGREE 8.
                 ; THE INPUT IS A 12-BYTE STRING POINTED
                   TO BY THE (HL)
                 ; REGISTER.  THE 8-BYTE OUTPUT STRING IS
5                  STORED AT (DE).
                 ; FOR MORE EFFICIENT MEMORY USAGE, (HL)
                   AND (DE) MAY BE
                 ; THE SAME BUFFER.
                 ; THIS ROUTINE MAKES USE OF A 16-BYTE
10                 WORKSPACE IN RAM,
                 ; CALLED HMATRIX.  THIS INTERMEDIATE
                   WORKSPACE IS USED TO
                 ; STORE DEMULTIPLEXED BUT UNCORRECTED
                   BYTES.
15               ;
                 ; THIS ROUTINE SHOULD BE CALLED WITH:
                 ; THE INPUT BUFFER POINTER IN THE (HL)
                   REGISTER,
```

```
1   CP/M MACRO ASSEM 2.0     #019  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
                     ; THE OUTPUT BUFFER POINTER IN THE (DE)
                       REGISTER, AND
                     ; THE NUMBER OF 12-BYTE BLOCKS TO DECODE
5                      IN THE A REGISTER
                     ;
                     ; THIS ROUTINE RETURNS WITH THE NEXT
                       INPUT BUFFER
                     ; ADDRESS IN THE (HL) REGISTER, THE
10                     NEXT OUTPUT BUFFER
                     ; ADDRESS IN THE (DE) REGISTER, AND
                       THE A REGISTER
                     ; CLEARED TO 0.
                     ;
15                   ; THIS ROUTINE STOMPS THE ALTERNATE
                       ACCUMULATOR AS WELL
                     ; AS THE B AND C REGISTERS.
                     ;
                     HAMCODE:EXAF          ;SAVE BLOCK COUNTER
20  2797+08
    2798 D5                 PUSH D        ;SAVE OUTPUT ADDRESS
                                           IN STACK
    2799 E5                 PUSH H        ;SAVE INPUT ADDRESS
                                           IN STACK
25  279A 21B640             LXI H,HMATRIX;CLEAR THE WORKSPACE
    279D 11B740             LXI D,HMATRIX+1
    27A0 010F00             LXI B,15
    27A3 3600               MVI M,0
                            LDIR
30  27A5+EDB0
    27A7 0E0C               MVI C,12      ;12 BITS TO UNSCRAMBLE
    27A9 D1                 POP D         ;INPUT POINTER TO
                                           (DE) NOW
    27AA 3AA540             LDA HBYTES    ;HOW MANY TO DEMUX?
35  27AD FE02               CPI 2         ;2 BYTE BLOCK?
```

```
                                JRNZ DLOOP    ;JUMP IF NOT
    27AF+2020
                          ;
                          ; THE FIRST STEP IS TO DEMULTIPLEX THE
                            INPUT BYTES.
                          ; THE FOLLOWING ROUTINE PUTS THE RESULT
                            OF THIS AT HMATRIX
                          ;
    27B1 21B640           DL:     LXI H,HMATRIX;THIS IS RESULT
                                                BUFFER
    27B4 3E04                     MVI A,4       ;CHECK FOR CORRECTION
                                                 FIELD
    27B6 B9                       CMP C
                                  JRC D19       ;JUMP IF NOT
    27B7+3801
    27B9 23                       INX H         ;ELSE POINT TO
                                                 CORRECTION BYTES
    27BA 79               D19:    MOV A,C       ;GET BIT COUNT
    27BB E603                     ANI 3         ;CHECK IF TIME FOR
                                                 NEW BYTE
                                  JRNZ D18      ;JUMP IF NOT
    27BD+2003
    27BF 1A                       LDAX D        ;GET THE NEW ONE
    27C0 13                       INX D
    27C1 47                       MOV B,A       ;NEW BYTE TO B
                          D18:    RALR B        ;SHIFT THE BITS OUT
    27C2+CB10
                                  RALR M        ;TO MEMORY
    27C4+CB16
    27C6 2323                     INX H ! INX H;DO 2ND ONE
                                  RALR B
    27C8+CB10
                                  RALR M
```

```
1   27CA+CB16
    27CC 0D                 DCR C          ;CHECK FOR DONE
                            JRNZ DL
    27CD+20E2
                            JMPR HAMM      ;THEN FINISH UP
5   27CF+1819
                    ;
    27D1 21B640     DLOOP:  LXI H,HMATRIX;RESET POINTER
```

```
1   CP/M MACRO ASSEM 2.0     #020  READ11 READER SOFTWARE
    10/23/80 AAJ (C) 1980 NLI
    27D4 3E04                MVI A,4       ;CORRECTION FIELD
                                            BITS?
    27D6 B9                  CMP C
5                            JRC D9        ;JUMP IF NOT
    27D7+3803
    27D9 21B740              LXI H,HMATRIX+1 ;ELSE RESET
                                               POINTER
    27DC 0608        D9:     MVI B,8       ;BIT COUNT TO B
10  27DE 1A                  LDAX D        ;GET THE NEXT BYTE
    27DF 13                  INX D         ;BUMP POINTER
    27E0 17          D2:     RAL           ;HI BIT TO CY
                             RALR M        ;TO MEMORY
    27E1+CB16
15  27E3 2323                INX H ! INX H ;TO NEXT OUTPUT BYTE
                             DJNZ D2       ;UNTIL BYTE FINISHED
    27E5+10F9
    27E7 0D                  DCR C         ;COUNT BIT NUMBER
                             JRNZ DLOOP    ;LOOP TILL BLOCK
20                                          DONE
    27E8+20E7
                     ;
    27EA EB          HAMM:   XCHG          ;INPUT POINTER TO
                                            HL
25  27EB E3                  XTHL          ;EXCHANGE FOR OUTPUT
                                            POINTER
    27EC EB                  XCHG          ;POINTER TO (DE)
    27ED 21B640              LXI H,HMATRIX ;DEMUXED CODEWORDS
                                             ARE HERE
30  27F0 3AA540              LDA HBYTES    ;GET NUMBER OF BYTES
                                            TO DE-MUX
    27F3 47                  MOV B,A       ;THIS COUNTER TO B
    27F4 4E          CORRECT:MOVE C,M      ;GET NEXT BYTE
                     ;
35
```

```
27F5 E5C5        HCODE1: PUSH H ! PUSH B ;SAVE USER
                                           REGISTERS
27F7 AF                  XRA A           ;CLEAR FOR HAMMING
                                          BITS
27F8 0608                MVI B,8         ;8 BITS/BYTE
27FA 212A28              LXI H,HTABLE    ;POINT TO ENCODING
                                          TABLE
                 HLOOP1: RALR C          ;HI BIT OF CHARACTER
                                          TO CY
27FD+CB11
                         JRNC NOBIT1     ;JUMP IF NO BIT
27FF+3001
2801 AE                  XRA M           ;RESIDUE ADD OF
                                          CODEWORD
2802 23          NOBIT1: INX H           ;POINT TO NEXT
                                          CODEWORD
                         DJNZ HLOOP1     ;ENCODE ALL PARITY
                                          BITS
2803+10F8
2805 C1E1                POP B ! POP H   ;RESTORE USER
                                           REGISTERS
                 ;
2807 23                  INX H           ;POINT TO CORRECTION
                                          FIELD
2808 AE                  XAR M           ;GET ERROR SYNDROME
                         JRZ NOERROR     ;JUMP IF DECODED
                                          CORRECTLY
2809+2812
280B E5                  PUSH H          ;SAVE INPUT POINTER
280C 213228              LXI H,SYNDROM   ;POINT TO CORRECTION
                                           TABLE
280F 85                  ADD L           ;ADD THE OFFSET
2810 6F                  MOV L,A
2811 3E00                MVI A,0
```

```
2813 8C                    ADC H
2814 67                    MOV H,A
2815 79                    MOV A,C          ;RECOVER THE
                                             MESSAGE BYTE
2816 AE                    XRA M            ;CORRECT THE ERROR
                                             BIT(S)
2817 4F                    MOV C,A          ;RETURN CORRECTED
                                             BYTE
2818 21A940                LXI H,ERCOUNT ;RECORD THE ERROR
281B 34                    INR M
281C E1                    POP H            ;RESTORE INPUT
                                             POINTER
281D 23          NOERROR:  INX H            ;NEXT INPUT BYTE
281E 79                    MOV A,C          ;MOVE GOOD BYTE
                                             TO OUTPUT
281F 12                    STAX D
```

```
1   CP/M MACRO ASSEM 2.0 #021     READ11 READER SOFTWARE
                                  10/23/80 AAJ (C) 1980 NLI
    2820 13          INX D
                     DJNZ CORRECT      ;UNTIL ALL DONE
    2821+10D1
5   2823 E1          POP H             ;RESTORE INPUT POINTER
                     EXAF              ;RECOVER GROUP NUMBER
    2824+08
    2825 3D          DCR A             ;COUNT GROUP
    2826 C29727      JNZ HAMCODE       ;LOOP TILL DONE
10  2829 C9          RET
                ;
    282A 0C0B0A0907HTABLE: DB 0CH,0BH,0AH,09H,07H,06H,05H,03H
    2832 0000000100SYNDROM:DB 0,0,0,1,0,2,4,8,41H,10H,20H,40H,
    80H,28H,18H,22H
15              ;
                ;
                ; SUBROUTINE CRLF SENDS A CARRIAGE RETURN
                ; AND A LINE FEED TO THE CONSOLE DEVICE.
                ;
20  2842 0E0D   CRLF:   MVI C,CR
    2844 CD692C         CALL CRT
    2847 0E0A           MVI C,LF
    2849 C3692C         JMP CRT
                ;
25              ;
    284C DB03   KBDSTAT:IN KBD
    284E 17             RAL
                        JRNC CHRWAIT  ;JUMP IF CHARACTER
                                      WAITING
30  284F+3002
    2851 AF             XRA A         ;ELSE RETURN W/ 0
    2852 C9             RET
    2853 AF     CHRWAIT:XRA A         ;SHOW CHARACTER WAITING
    2854 3D             DCR A
35  2855 C9             RET
                ;
    2856 CD4C28 KBDWAIT:CALL KBDSTAT  ;WAIT FOR INPUT
                        JRZ KBDWAIT   ;THEN FALL THRU
```

```
2859+28FB
                    ;
285B CDB528         KBDIM:  CALL STIME      ;WAIT FOR 3 MS TO LET
                                            BITS SETTLE
285E DB03                   IN KBD          ;THEN GET THE CHARACTER
2860 E67F                   ANI 07FH        ;DROP THE HI BIT
2862 C9                     RET
                    ;
                    ;
                    ;
                    ; STEPPER MOTOR CONTROLLER ROUTINES
                    ;
                    ; THESE ROUTINES ALLOW STEPPER MOTOR CONTROL
                    ;       OUTTRK  - STEP OUT ONE TRACK DISTANCE
                    ;       INTRK   - STEP IN ONE TRACK DISTANCE
                    ;
                    ;
                    SKEWIN: INRLY           ;INCREMENT SKEW COUNT
2863+FD
2864+2C
                    IF HALFSTP
                            CALL DOSTEP
                            INRLY
                    ENDIF
                            JMPR DOSTEP     ;STEP IT
```

```
1   CP/M MACRO ASSEM 2.0   #022   READ11 READER SOFTWARE
                                   10/23/80 AAJ (C) 1980 NLI

    2865+180E
                SKEWOUT:DCRLY          ;DECREMENT SKEY COUNT
    2867+FD
5   2868+2D
                IF HALFSTP
                    CALL DOSTEP
                    INRLY
                ENDIF
10                  JMPR DOSTEP
    2869+180A
                OUTONE: INRHY          ;INCREMENT CARRIAGE
                                       COUNT
    286B+FD
15  286C+24
                    DCXIX              ;DECREMENT CARRIAGE
                                       POSITION
    286D+DD2B
                    JMPR DOSTEP
20  286F+1804
                INONE: DCRHY           ;DECREMENT CARRIAGE
                                       COUNT
    2871+FD
    2872+25
25                  INXIX              ;INCREMENT CARRIAGE
                                       POSITION
    2873+DD23
                ;
                DOSTEP: EXAF           ;SAVE THE ACCUMULATOR
30  2875+08
    2876 CDB528     CALL STIME         ;WAIT FOR THE STEPPER
                    MOVALY             ;GET THE SKEW COUNT
    2879+FD
    287A+7D
35  287B CD8F28     CALL SBYTE         ;GET THE HI NYBBLE CODE
    287E 8787       ADD A ! ADD A      ;*16
    2880 8787       ADD A ! ADD A
                    EXX                ;STORE IN ALT D
```

```
2882+D9
2883 57          MOV D,A
                 MOVAHY          ;GET CARRIAGE COUNT
2884+FD
2885+7C
2886 CD8F28      CALL SBYTE      ;GET THE LO NYBBLE
2889 B2      SNYBLE: ORA D       ;ADD THE HI NYBBLE
                 EXX             ;GET ORIG REGISTERS
                                 BACK
288A+D9
288B D300        OUT STEPPER     ;SEND TO THE MOTORS
                 EXAF            ;RECOVER ACCUMULATOR
288D+08
288E C9          RET
                 ;
                 SBYTE:
                 IF (NOT HALFSTP)
288F E603        ANI 3           ;MOD 4 FOR NORMAL STEP
                 ENDIF
                 IF HALFSTP
                 ANI 7           ;MOD 8 FOR HALF STEP
                 ENDIF
                 JRNZ S1         ;JUMP IF NOT STEP 0
2891+2003
2893 3E0AC9      MVI A,STEPO ! RET
2896 3D      S1: DCR A ! JRNZ S2
2897+2003
2899 3E06C9      MVI A,STEP1 ! RET
```

```
1   CP/M MACRO ASSEM 2.0  #023  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
    289C 3D       S2: DCR A ! JRNZ S3
    289D+2003
    289F 3E05C9       MVI A,STEP2 ! RET
5               S3:
                IF HALFSTP
                    DCR A ! JRNZ S4
                ENDIF
    28A2 3E09C9       MVI A,STEP3 ! RET
10              ;
                IF HALFSTP
                ;
                S4: DCR A ! JRNZ S5
                    MVI A, STEP4 ! RET
15              S5: DCR A ! JRNZ S6
                    MVI A,STEP5 ! RET
                S6: DCR A ! JRNZ S7
                    MVI A,STEP6 ! RET
                S7: MVI A,STEP7 ! RET
20              ;
                ENDIF
                ;
    28A5 0604   INTRK: MVI B,TRACK    ;MOVE ONE TRACK DIS-
                                      TANCE
25  28A7 CD7128 STEPIN: CALL INONE    ;WE STEP CARRIAGE IN
                    DJNZ STEPIN       ;UNTIL FINISHED
    28AA+10FB
    28AC C9         RET
                ;
30  28AD 0604   OUTTRK: MVI B,TRACK   ;MOVE ONE TRACK DIS-
                                      TANCE
    28AF CD6B28 STEPOUT:CALL OUTONE   ;WE STEP CARRIAGE OUT
                    DJNZ STEPOUT      ;UNTIL FINISHED
    28B2+10FB
35  28B4 C9         RET
                ;
                ; SUBROUTINE STIME SETS THE STEPPER MOTOR
                CLOCK DELAY.
```

```
                    ;
28B5 F5     STIME: PUSH PSW      ;SAVE THE ACCUMULATOR
28B6 E5            PUSH H        ;AND THE HL
28B7 219501        LXI H,STIMER  ;TIME COUNTER TO HL
28BA 2B     WAIT1: DCX H         ;COUNT THE TIME
28BB 7C            MOV A,H       ;WATCH FOR 0
28BC B5            ORA L
                   JRNZ WAIT1    ;LOOP TILL TIME UP
28BD+20FB
28BF E1            POP H         ;RECOVER HL
28C0 F1            POP PSW       ;RECOVER ACCUMULATOR
28C1 C9            RET
                   ;
                   ;
                   ; SUBROUTINE EJECT
                   ;
                   ; SUBROUTINE EJECT CAUSES THE FILM TO BE
                   HOMED TO
                   ; THE FULLY "OUT" POSITION.
                   ;
28C2 010407 EJECT: LXI B,STEPS+20 ;COUNT TRACKS HERE
28C5 DB01   EJ1: IN PORT         ;WATCH FOR SENSOR
28C7 17            RAL
                   JRC EJ2       ;CONTINUE IF NOT DONE
28C8+3806
                   LXIX 0        ;CLEAR CARRIAGE POSI-
                                 TION REGISTER
28CA+DD21
```

```
CP/M MACRO ASSEM 2.0    #024   READ11 READER SOFTWARE
10/23/80 AAJ (C) 1980 NLI
 28CC+000
 28CE AF                XRA A
 28CF C9                RET
 28D0 CD6B28    EJ2:    CALL OUTONE      ;STEP OUT 1 TRACK
 28D3 0B                DCX B            ;COUNT TRACK #
 28D4 78B1              MOV A,B ! ORA C
                        JRNZ EJ1         ;TILL DONE
 28D6+20ED
 28D8 AF3D              XRA A ! DCR A    ;SHOW BAD
 28DA C9                RET
                ;
                ;
                ; SUBROUTINE HOME
                ;
                ; SUBROUTINE HOME CAUSES THE FILM TO BE
                  HOMED TO
                ; THE FULLY "IN" POSITION
                ;
 28DB E5        HOME:   PUSH H           ;SAVE USER HL
 28DC 01F006            LXI B, STEPS     ;COUNT STEPS HERE
                        PUSHIX           ;GET THE IX TO HL
 28DF+DDE5
 28E1 E1                POP H
 28E2 79        HOME1:  MOV A,C          ;CHECK FOR DONE
 28E3 95                SUB L            ;LO BYTE MATCH?
                        JRNZ HOME2       ;CONTINUE IF NOT
 28E4+2006
 28E6 7894              MOV A,B ! SUB H  ;HOW ABOUT HI MATCH?
                        JRNZ HOME2       ;CONTINUE IF NOT
 28E8+2002
 29EA E1                POP H            ;RESTORE H
 28EB C9                RET
 28EC CD7128    HOME2   CALL INONE       ;STEP IN ONE STEP
```

```
                28EF 23          INX H          ;COUNT IT
                                 JMPR HOME1     ;LOOP TILL DONE
                28F0+18F0
                                 ;
                                 ; SUBROUTINE CENTER INITS THE CARRIAGE
                                   POSITION
                                 ; TO APPROX. TRACK 128 AND CENTERS THE
                                   SKEW MOTOR
                                 ; TO THE CENTER OF ITS TRAVEL
                                 ;
                28F2 CDC228      CENTER: CALL EJECT     ;FULLY OUT
                                                        POSITION
                28F5 015E02              LXI B,FMARGIN+(64*TRACK) ;CARRIAGE
                                                                  DESTINATION
                28F8 11EC00              LXI D,SKLIMIT+20.;SKEW LIMIT
                28FB 78B1        CLOOP:  MOV A,B ! ORA C ;CHECK IF DONE
                                                        WITH CARRIAGE
                                         JRNZ SK3       ;JUMP IF NOT DONE
                                                        YET
                28FD+2007
                28FF 7A                  MOV A,D        ;CHECK SKEW ALSO
                                         REX 7,A        ;CLEAR DIRECTION
                                                        FLAG FOR TEST
                2900+CBBF
                2902 B3                  ORA E
                2903 C8                  RZ             ;RETURN IF BOTH
                                                        DONE
                                         JMPR CDONE     ;ELSE FINISH SKEW
                2904+1803
                2906 0B          SK3:    DCX B          ;COUNT THE
                                                        CARRIAGE STEP
                                         DCRHY          ;BUMP THE STEPPER
                                                        DRIVE COUNT
                2907+FD
                2908+25
                                 CDONE:  BIT 7,D        ;ARE WE CENTERING
                                                        THE SKEW?
```

```
1     2909+CB7A
                        JRZ SK1         ;JUMP IF STILL
                                        PINNING IT
```

```
CP/M MACRO ASSEM 2.0  #025 READ11 READER SOFTWARE
                            10/23/80 AAJ (C) 1980 NLI

290B+280F
290D 7B          MOV A,E            ;CHECK FOR DONE
290E B7          ORA A
                 JRNZ CLOOP1        ;JUMP IF SKEW NOT DONE
290F+2005
2911 CD7528      CALL DOSTEP        ;DO THE CARRIAGE STEP
                 JMPR CLOOP         ;AND LOOP TILL COMPLETE
2914+18E5
2916 CD6328  CLOOP1: CALL SKEWIN    ;DO CARRIAGE PLUS SKEW
2919 1D          DCR E              ;DECREMENT THE COUNTER
                 JMPR CLOOP         ;TILL DONE
291A+18DF
291C 1B      SK1:DCX D              ;COUNT THE PIN DIREC-
                                    TION COUNTER
291D 7AB3        MOV A,D ! ORA E    ;CHECK FOR PIN CYCLE
                                    DONE
                 JRZ SK2            ;JUMP IF FIRST HALF
                                    DONE
291F+2805
2921 CD6728      CALL SKEWOUT       ;MOVE THE SKEW TO
                                    PINNED POSITION
                 JMPR CLOOP         ;LOOP TILL COMPLETE
2924+18D5
             SK2:SETB 7,D           ;SET 2ND HALF FLAG
2926+CBFA
2928 1E6C        MVI E,SKLIMIT/2    ;CENTERING COUNT TO E
                 JMPR CLOOP         ;LOOP TILL DONE
292A+18CF
                 ;
                 ;
                 ; SUBROUTINE GETCUR
                 ;
                 ; SUBROUTINE GETCUR FETCHES, DECODES, AND
                 ERROR
                 ; CORRECTS THE CURRENT LINE (LINE POINTED TO
                 BY CURLINE).
                 ;
```

```
                ; IF THE INITIAL TRY AT DECODING IS UN-
                SUCCESSFUL, THE
                ; ROUTINE WILL TRY A NUMBER OF DIFFERENT
                APPROACHES:
                ; IT WILL TRY +-1 and +-2 STEPS FROM CENTER,
                AND ALSO
                ; RE-SKEW ADJUSTING THE FILM FOR THE CURRENT
                TRACK.
                ;
                ; IF SUCCESSFUL, THIS ROUTINE RETURNS WITH
                THE LINE
                ; AT OUTBUF AND THE CY FLAG RESET AND 0 IN
                THE A REGISTER.
                ; IF UNSUCCESSFUL, IT WILL RETURN WITH CY
                SET AND
                ; NON-ZERO IN THE A REGISTER. OUTBUF WILL BE
                STOMPED.
                ;
292C AF         GETCUR: XRA A           ;RESET THE SAFETY VALVE
292D 32A740         STA FUSE
2930 CD4029         CALL GC1            ;TRY WITH NO SKEW
2933 C8             RZ
2934 CD512B         CALL SKEW2          ;ONLY IF WE HAVE TO
2937 CD412C         CALL CEFFECT
293A CD512B         CALL SKEW2
293D CD412C         CALL CEFFECT
2940 AF         GC1:XRA A               ;RESET TRY NUMBER
                                        COUNTER
2941 32A440         STA TRYNO
2944 CD8225     WHERE: CALL INTRAK      ;GET THE ADDRESS
2947 3AA740         LDA FUSE            ;CHECK FOR GARBAGE LOOP
294A 3C             INR A
294B 32A740         STA FUSE
294E FE10           CPI 16              ;ARE WE SICK?
                    JRNZ WARE           ;JUMP IF NOT
2950+2004
2952 AF3D           XRA A ! DCR A       ;SHOW BAD
2954 37             STC
```

```
1   CP/M MACRO ASSEM 2.0  #026  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

    2955 C9           RET
    2956 CD8F26  WARE: CALL GETFTA      ;
                      JRNC CURSE        ;JUMP IF ADDRESS FOUND
5   2959+3005
    295B CD7128       CALL INONE        ;STEP CARRIAGE
                      JMPR WHERE        ;UNTIL TRACK ID FOUND
    295E+18E4
                  ;
10  2960 2AB040  CURSE: LHLD CURLINE   ;WE SHOULD BE HERE
    2963 95           SUB L             ;OK? (OFFSET TO A)
                      JRZ DEC1          ;DECODE PIXELS IF OK
    2964+2832
    2966 F27429       JP FAROUT         ;JUMP IF TOO FAR OUT
15                    NEG               ;MAKE OFFSET POSITIVE
    2969+ED44
    296B 4F           MOV C,A           ;COUNTER TO C
    296C CDA528  GET1:CALL INTRK        ;GO OUT ONE TRACK DIS-
                                        TANCE
20  296F 0D           DCR C             ;COUNT OFFSET IN C
                                        REGISTER
                      JRNZ GET1         ;LOOP UNTIL AT PROPER
                                        TRACK
    2970+20FA
25                    JMPR GC1          ;TRY AGAIN AT PROPER
                                        POSITION
    2972+18CC
    2974 4F      FAROUT: MOV C,A        ;COUNTER TO C
    2975 CDAD28  GET2: CALL OUTTRK      ;GO IN ONE TRACK DIS-
30                                      TANCE
    2978 0D           DCR C
                      JRNZ GET2         ;UNTIL THERE
    2979+20FA
                      JMPR GC1          ;THEN RETRY AT PROPER
35                                      PLACE
    297B+18C3
                  ;
    297D 214040  DECODE: LXI H,OUTBUF   ;THIS IS OUTPUT POINTER
```

```
2980 E5          PUSH H              ;SAVE IN STACK
2981 2AAC40      LHLD LSTART         ;GET START OF LINE
2984 0E63        MVI C,(LLENGTH AND OFFH)+3 ;LINE LENGTH
                                     COUNTER TO C
2986 0608    DX: MVI B,8             ;8 BITS/BYTE
2988 CD3126  D1: CALL GETBIT         ;GET NEXT BIT
                 RALR E              ;PUT BYTE IN E
298B+CB13
                 DJNZ D1             ;UNTIL DONE
298D+10F9
298F E3          XTHL                ;GET OUTPUT POINTER
2990 73          MOV M,E             ;COMPLETE BYTE TO
                                     MEMORY
2991 23          INX H               ;POINT TO NEXT POSITION
2992 E3          XTHL                ;PUT POINTER BACK IN
                                     STACK
2993 0D          DCR C               ;COUNT BYTE #
                 JRNZ DX             ;LOOP TILL WHOLE LINE
                                     DECODED
2994+20F0
2996 E1          POP H               ;FIX STACK
2997 C9          RET                 ;DONE
                 ;
2998 CD7D29  DEC1: CALL DECODE       ;GET THE PIXELS
                 ;
                 IF HAMMING
299B 214040      LXI H,OUTBUF        ;POINT TO DECODED
                                     STRING
299E 545D        MOV D,H ! MOV E,L   ; COPY TO DE
29A0 3E08        MVI A,8             ;8 BYTE GROUPS
29A2 32A540      STA HBYTES
29A5 AF          XRA A
29A6 32A940      STA ERCOUNT         ;RESET HAMMING COUNT
29A9 3E08        MVI A,GROUPS        ;# OF HAMMING GROUPS TO
                                     A
```

```
1   CP/M MACRO ASSEM 2.0  #027  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
    29AB CD9727       CALL HAMCODE      ;ERROR CORRECT STRING
                  ENDIF
    29AE 3E02         MVI A,2           ;CRC IS 2 BYTE GROUP
5   29B0 32A540       STA HBYTES
    29B3 3E01         MVI A,1           ;ONE HAMMING GROUP
    29B5 CD9727       CALL HAMCODE      ;DECODE IT
    29B8 2A8040       LHLD OUTBUF+(GROUPS*8) ;GET THE CRC WORD
    29BB EB           XCHG              ;CRC IS IN DE
10                    EXX               ;GET ALTERNATES
    29BC+D9
    29BD 210000       LXI H,0           ;CLEAR CRC REGISTER
                      EXX               ;PUT THEM BACK
    29C0+D9
15                IF HAMMING
    29C1 0640         MVI B,(GROUPS*8)  ;#OF RESULTANT BYTES IN
                                        B
                  ELSE
                      MVI B,LLENGTH     ;THIS MANY BYTES
20                ENDIF
    29C3 214040       LXI H,OUTBUF      ;POINT TO THE BYTES
    29C6 7E       CRCLOOP:MOV A,M       ;GET NEXT BYTE
    29C7 23           INX H             ;BUMP COUNTER
                  ;
25                ; CRC SUBROUTINE
                  ;
                  ; THIS SUBROUTINE COMPUTES THE 16-BIT CYCLIC
                  REDUNDANCY
                  ; CHECK WORD USING THE STANDARD CCITT POLY-
30                NOMIAL:
                  ;
                  ; X^16 + X^15 + X^13 + X^7 + X^4 + X^2 +
                  X + 1
                  ;
35                ; THE INPUT BYTE IS PASSED IN THE A REGISTER,
                  AND THE
                  ; NEW OUTPUT WORD IS STORED IN THE ALTERNATE
                  H REGISTER.
```

```
1                   ;
                    CRC:EXX                 ;GET ALTERNATES
    29C8+D9
    29C9 4F             MOV C,A             ;SAVE NEW BYTE IN
                                            ALTERNATE C
5   29CA 7C             MOV A,H             ;LO BYTE OF CRC
    29CB E680           ANI 10000000B       ;MASK FOR HI BIT (Q BIT)
                        EXAF                ;SAVE STATUS IN ALTER-
                                            NATE F
    29CD+08
10  29CE 29             DAD H               ;2 * R(X)
    29CF 79             MOV A,C             ;RECOVER THE NEW BYTE
    29D0 85             ADD L               ;ADD TO SHIFTED LO BYTE
    29D1 6F             MOV L,A             ;RESTORE NEW BYTE
                        EXAF                ;RECOVER STATUS
15  29D2+08
                        JRZ QB2             ;JUMP IF Q BIT WAS ZERO
    29D3+2808
    29D5 7C         QB: MOV A,H             ;DO POLYNOMIAL IF Q BIT
                                            IS ONE
20  29D6 EEA0           XRI 10100000B       ;APPLY MSB OF POLY-
                                            NOMIAL
    29D8 67             MOV H,A             ;RESTORE
    29D9 7D             MOV A,L             ;THEN APPLY LO BYTE
    29DA EE97           XRI 10010111B
25  29DC 6F             MOV L,A             ;RESTORE
                    QB2:EXX                 ;STORE CRC WORD & RE-
                                            STORE REGISTERS
    29DD+D9
                    ;
30                  DJNZ CRCLOOP            ;CRC ON WHOLE LINE
    29DE+10E6
                        EXX                 ;RECOVER CRC REGISTER
    29E0+D9
```

```
1   CP/M MACRO ASSEM 2.0  #028  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
    29E1 E5          PUSH H          ;SAVE IN STACK
                     EXX             ;GET ORIGS BACK
    29E2+D9
5   29E3 E1          POP H           ;CALCULATED CRC IN HL,
                                     ;RECEIVED CRC IN DE
    29E4 AF          XRA A           ;CLEAR THE CY BIT
                     DSBC D          ;COMPARE THEM
    29E5+ED52
10  29E7 7CB5        MOV A,H ! ORA L ;ARE THEY THE SAME?
    29E9 C8          RZ              ;RETURN IF GOOD
    29EA 21A440  OHSHIT: LXI H,TRYNO ;POINT TO TRY NUMBER
                                     COUNTER
    29ED 7E          MOV A,M         ;GET THIS VALUE
15  29EE 34          INR M           ;UPDATE COUNTER TO NEXT
                                     VALUE
    29EF B7          ORA A           ;CHECK FOR 1ST PASS
                     JRNZ BAD1       ;JUMP IF NOT 1ST TIME
                                     THRU
20  29F0+2008
    29F2 CD6B28      CALL OUTONE     ;MOVE OUT 1
    29F5 CD2A2A      CALL TRYIT      ;CHECK OUT THE LINE
                     JMPR BAD1A      ;GO TO NEXT TRY IF NO
                                     GOOD
25  29F8+1803
    29FA 3D      BAD1:DCR A          ;CHECK FOR TRY # 2
                     JRNZ BAD2       ;JUMP IF NOT
    29FB+200A
    29FD 0602    BAD1A:MVI B,2       ;MOVE IN 2 (NET IN ONE)
30  29FF CDA728      CALL STEPIN
    2A02 CD2A2A      CALL TRYIT      ;TRY AT ONE STEP IN
                     JMPR BAD2A      ;DO NEXT RETRY IF BAD
    2A05+1803
    2A07 3D      BAD2:DCR A          ;CHECK FOR PASS #3
35                   JRNZ BAD3       ;JUMP IF NOT PASS 3
    2A08+200A
    2A0A 0603    BAD2A;MVI B,3       ;3 STEPS OUT (NET OUT
                                     2)
```

```
2A0C CDAF28      CALL STEPOUT      ;GO THERE
2A0F CD2A2A      CALL TRYIT        ;TRY AT TWO STEPS OUT
                 JMPR BAD3A        ;JUMP IF THIS TRY BAD
2A12+180C
2A14 3D      BAD3:DCR A            ;TRY FOR 4TH PASS
                 JRZ BAD3A         ;JUMP IF 4TH PASS
2A15+2809
2A17 0602    NOWAY:MVI B,2         ;CENTER THE CARRIAGE
                                   BACK UP
2A19 CDAF28      CALL STEPOUT      ;TO MIDDLE OF TRACK
2A1C AF3D        XRA A ! DCR A     ;SHOW NON-ZERO
2A1E 37          STC
2A1F C9          RET               ;RETURN WITH NOTHING
2A20 0604    BAD3A:MVI B,4         ;4 STEPS IN(NET IN 2)
2A22 CDA728      CALL STEPIN       ;GO THERE
2A25 CD2A2A      CALL TRYIT        ;CHECK FOR GOOD LINE
                 JMPR NOWAY        ;KICK OUT IF NOT GOOD
2A28+18ED
                 ;
2A2A CD8225  TRYIT:CALL INTRAK     ;GET THE PIXELS
2A2D CD8F26      CALL GETFTA       ;TRY FOR SYNC
                 JRNC GOODFTA      ;JUMP IF GOOD FTA IS
                                   FOUND
2A30+3001
2A32 C9          RET               ;CONTINUE
                 ;
2A33 F1      GOODFTA:POP PSW       ;FIX THE STACK
2A34 C34429      JMP WHERE         ;TRY THE LINE HERE
                 ;
                 ;
```

```
1   CP/M MACRO ASSEM 2.0  #029  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

                    ;
                    ; SUBROUTINE GOLINE
                    ;
5                   ; SUBROUTINE GOLINE GOES TO THE TRACK POINTED
                    TO
                    ; BY THE (DE) REGISTER PAIR.
                    ; IF SUCCESSFUL, IT RETURNS WITH 0 IN THE A
                    REGISTER
10                  ; AND THE CY FLAG RESET.
                    ; IF UNSUCCESSFUL(BAD TRACK NUMBER) IT RE-
                    TURNS WITH
                    ; THE CY FLAG SET AND NON-ZERO IN THE A
                    REGISTER.
15                  ;
    2A37 217701  GOLINE:LXI H,1500/TRACK ;MAX NUMBER OF LEGAL
                                          TRACKS
    2A3A AF          XRA A               ;CLEAR CY ...
                     DSBC D              ;FOR SUBTRACT
20  2A3B+ED52
                     BIT 7,H             ;DID WE GO NEGATIVE?
    2A3D+CB7C
                     JRZ GOODADD         ;JUMP IF NOT
    2A3F+2802
25  2A41 3D          DCR A               ;SHOW BAD
    2A42 C9          RET                 ;AND RETURN
    2A43 2AB040  GOODADD:LHLD CURLINE    ;GET CURRENT LOCATION
    2A46 EB          XCHG                ;CURRENT LOC. IN DE,
                                         DESIRED IN HL
30  2A47 22B040      SHLD CURLINE        ;SAVE DESIRED LOCATION
                                         AT CURLINE
    2A4A EB          XCHG                ;SWAP BACK.
    2A4B AF          XRA A               ;CLEAR THE CARRY BIT
                                         FOR....
35                   DSBC D              ;16 BIT SUBTRACT
    2A4C+ED52
                     BIT 7,H             ;ARE WE TOO CLOSE TO
                                         TRACK 0?
```

```
2A4E+CB7C
                    JRNZ GO2            ;STEP OUT IF TOO CLOSE
2A50+2009
2A52 7CB5    GO1:MOV A,H ! ORA L        ;CHECK IF WE ARE THERE
2A54 C8          RZ                     ;DONE IF SO
2A55 CDAD28      CALL OUTTRK            ;STEP IN ONE TRACK DIS-
                                        TANCE
2A58 2B          DCX H                  ;COUNT THE STEP
                 JMPR GO1               ;LOOP TILL DONE
2A59+18F7
2A5B 7D      GO2:MOV A,L                ;NEGATE HL PAIR
2A5C 2F          CMA                    ;BY COMPLEMENT...
2A5D 6F          MOV L,A
2A5E 7C          MOV A,H
2A5F 2F          CMA
2A60 67          MOV H,A
2A61 23          INX H                  ;AND INCREMENT
2A62 7CB5    GO3:MOV A,H ! ORA L        ;CHECK FOR DONE
2A64 C8          RZ                     ;RETURN IF FINISHED
2A65 CDA528      CALL INTRK             ;GO OUT ONE TRACK DIS-
                                        TANCE
2A68 2B          DCX H
                 JMPR GO3               ;UNTIL WE ARE THERE
2A69+18F7
                 ;
                 ;
                 ; SUBROUTINE GETNUM
                 ;
                 ; SUBROUTINE GETNUM SEEKS TO A NUMBERED
                 LINE POINTED
                 ; TO BE THE (DE) PAIR AND RETRIEVES IT TO
                 OUTBUF.
                 ; IF SUCCESSFUL, IT RETURNS WITH THE DATA AT
                 OUTBUT,
                 ; THE CY FLAG RESET AND 0 IN THE A REGISTER.
                 ; IF UNSUCCESSFUL, IT RETURNS WITH CY SET,
                 NON-ZERO
                 ; IN THE A REGISTER AND GARBAGE AT OUTBUF.
```

```
1   CP/M MACRO ASSEM 2.0  #030  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

                    ;
    2A6B CD372A  GETNUM:CALL GOLINE     ;SEEK TO THE LINE
    2A6E C32C29      JMP GETCUR         ;AND RETRIEVE IT
5                   ;
                    ;
                    ; SUBROUTINE GETNXT
                    ;
                    ; SUBROUTINE GETNXT RETRIEVES THE NEXT TRACK
10                  ON THE FILM.
                    ;
                    ; IF SUCCESSFUL, IT RETURNS WITH THE CY FLAG
                    RESET,
                    ; 0 IN THE A REGISTER, AND THE TRACK AT
15                  OUTBUF.
                    ; IF UNSUCCESSFUL, IT RETURNS WITH THE CY
                    FLAG SET,
                    ; NON-ZERO IN THE A REGISTER AND GARBAGE AT
                    OUTBUF.
20                  ;
    2A71 2AB040  GETNXT:LHLD CURLINE    ;GET THE LINE NUMBER
    2A74 23          INX H              ;GO TO NEXT
    2A75 EB          XCHG               ;TO DE
    2A76 CD372A      CALL GOLINE        ;GO THERE
25  2A79 C32C29      JMP GETCUR         ;AND GET IT
                    ;
                    ;
                    ;
                    ; SUBROUTINE SKEW
30                  ;
                    ; SUBROUTINE SKEW IS USED TO ROTATE THE FILM
                    ; SO THAT THE TRACKS ARE PARALLEL WITH THE
                    ; ROTATION OF THE READ HEAD, IF SUCCESSFUL,
                    IT
35                  ; RETURNS WITH THE FILM POSITIONED OVER TRACK
                    ; 0 THE ROTATION ADJUSTED, THE A REGISTER
                    ; IS 0 AND THE CY FLAG IS CLEARED.
                    ;
```

```
                    ; IF UNSUCCESSFUL, THE A REGISTER CONTAINS A
                    ; NON-ZERO AND THE CY FLAG IS SET.
                    ;
                    ; THE ROUTINE RUNS IN 4 PARTS:
                    ;
                    ; 1) THE CARRIAGE IS EJECTED, THE SKEW IS
                    CENTERED,
                    ;   AND THE CARRIAGE IS ADVANCED TO APPROX.
                    ;   TRACK 128.
                    ;
                    ; 2) ROUGH SKEW ADJUSTMENT IS DONE AT APPROX
                    TRACK
                    ;   128, AND THE CARRIAGE IS ADVANCED TO
                    APPROX.
                    ;   TRACK 10.
                    ;
                    ; 3) FINE SKEW IS DONE AT APPROX TRACK 10 AND
                    THE
                    ;   FILM IS ADVANCED TO TRACK 1.
                    ;
                    ; 4) THE FILM IS ADVANCED A STEP AT A TIME
                    TO TRACK
                    ;   0, AND FINAL SKEW AND CARRIAGE ADJUSTMENT
                    ;   IS DONE.
                    ;
2A7C CDF228         SKEW:CALL CENTER        ;INIT THE STEPPERS
2A7F 21182E             LXI H,SKTABLE       ;POINT TO MIDDLE OF
                                            TABLE
2A82 22B240             SHLD SKADDR         ;STORE THE TABLE
                                            POINTER
2A85 21A400             LXI H,(8*16)+00100100B ;INIT THE BIT
                                            COUNTER
2A88 22B440             SHLD BITBYTE        ;STORE THIS TOO
2A8B 210000             LXI H,0             ;RESET THE TRACK
                                            COUNTER
2A8E 22B040             SHLD CURLINE
                    ;
2A91 CD512B             CALL SKEW2          ;ROUGH SKEW AT TRACK 64
```

```
1   CP/M MACRO ASSEM 2.0  #031  READ11 READER SOFTWARE
                                 10/23/80 AAJ (C) 1980 NLI
    2A94 C0          RNZ              ;RETURN IF SKEW IM-
                                      POSSIBLE
    2A95 7C          MOV A,H          ;FTA IN A
5   2A96 D60A        SUI 10           ;LESS 10 TRACKS
    2A98 5F1600      MOV E,A ! MVI D,0 ;ADDRESS TO DE
    2A9B 210000      LXI H,0          ;CLEAR RESULT REGISTER
    2A9E 0604        MVI B,TRACK      ;TRACK WIDTH TO B
    2AA0 19      SK5:DAD D            ;ADD TRACKS
10                   DJNZ SK5         ;UNTIL TOTAL STEPS IN
                                      HL
    2AA1+10FD
    2AA3 59          MOV E,C          ;PLUS CARRIAGE OFFSET
                     BIT 7,E          ;IS IT NEGATIVE?
15  2AA4+CB7B
                     JRZ POS          ;JUMP IF NOT
    2AA6+2801
    2AA8 15          DCR D            ;TURN HI BYTE NEGATIVE
    2AA9 19      POS:DAD D            ;ADD THE CARRIAGE OFF-
20                                    SET
    2AAA CD6B28  SK6:CALL OUTONE      ;GOTO TRACK 10
    2AAD 2B          DCX H            ;COUNT THE STEPS
    2AAE 7CB5        MOV A,H ! ORA L  ;CHECK FOR DONE
                     JRNZ SK6         ;LOOP TILL THERE
25  2AB0+20FB
                 ;
    2AB2 CD512B      CALL SKEW2       ;FINE SKEW AT TRACK 10
    2AB5 C0          RNZ              ;RETURN IF SKEW NOT
                                      SUCCESSFUL
30  2AB6 AF          XRA A            ;CLEAR RESULT REGISTER
    2AB7 25          DCR H            ;TRACK 1 DISTANCE LESS
                                      1
    2AB8 0604        MVI B,TRACK      ;TURN INTO STEPS
    2ABA 84      SK0:ADD H            ;ADD TO GET TOTAL STEPS
35                   DJNZ SK0
    2ABB+10FD
    2ABD 3D          DCR A            ;LESS ONE STEP
    2ABE 47          MOV B,A          ;COUNTER TO B
```

```
2ABF CDAF28       CALL STEPOUT      ;GOTO TRACK 1
2AC2 110000       LXI D,0           ;D IS FTA,E IS BTA
2AC5 CD6B28  SK7: CALL OUTONE       ;STEP ONE CARRIAGE
                                    STEP
2AC8 CD8225       CALL INTRAK       ;GET PIXELS
2ACB 7A           MOV A,D           ;CHECK FOR FOUND AL-
                                    READY
2ACC B7           ORA A
                  JRZ SK10          ;JUMP IF NOT
2ACD+2803
2ACF 14           INR D             ;BUMP OFFSET
                  JMPR NOFTA        ;LOOP FOR BTA
2AD0+1809
2AD2 CD8F26  SK10:CALL GETFTA       ;TRY FOR FTA
                  JRC NOFTA         ;JUMP IF NOT THERE
2AD5+3804
2AD7 B7           ORA A             ;PROPER TRACK ?
                  JRNZ NOFTA        ;JUMP IF NOT TRACK 0
2AD8+2001
2ADA 14           INR D             ;SHOW WE GOT IT
2ADB 7B      NOFTA:MOV A,E          ;CHECK IF WE GOT IT
                                    ALREADY
2ADC B7           ORA A
                  JRZ SK11          ;JUMP IF NOT FOUND YET
2ADD+2803
2ADF 1C           INR E             ;ADD OFFSET IF FOUND
                  JMPR SK8          ;LOOP FOR FTA
2AE0+1809
2AE2 CD0F27  SK11:CALL GETBTA       ;TRY FOR BACK TRACK
                                    ADDRESS
                  JRC SK7           ;JUMP IF NOT FOUND
2AE5+38DE
```

```
1   CP/M MACRO ASSEM 2.0  #032  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
    2AE7 B7          ORA A              ;CHECK FOR TRACK 0
                     JRNZ SK7           ;JUMP IF NOT TRACK 0
    2AE8+20DB
5   2AEA 1C          INR E              ;SHOW WE FOUND IT
    2AEB 7A      SK8:MOV A,D            ;CKECK IF WE GOT FTA
                                        TOO
    2AEC B7          ORA A
                     JRZ SK7            ;JUMP IF WE NEED FTA
10                                      ALSO
    2AED+28D6
                                        ;ELSE FALL THRU IF BOTH
                                        FOUND
    2AEF 7A          MOV A,D            ;FTA TO A
15  2AF0 93          SUB E              ;OFFSET TO A
                     JRZ SKDONE         ;JUMP IF PERFECT
    2AF1+280F
    2AF3 FA382B      JM SK9             ;JUMP IF OFF CCW
    2AF6 47          MOV B,A            ;SAVE THE BYTE
20  2AF7 AF          XRA A              ;CLEAR CY TO SHOW CW
                                        ROTATION
    2AF8 78          MOV A,B            ;GET BYTE BACK
    2AF9 CDD82B  CW: CALL LOOKUP        ;GET THE SKEW STEPS
    2AFC 47          MOV B,A            ;SKEW OFFSET TO B
25  2AFD CD6328  SK12:CALL SKEWIN       ;ADJUST SKEW
                     DJNZ SK12          ;UNTIL CENTERED
    2B00+10FB
    2B02 0603    SKDONE:MVI B,3         ;3 PASSES AT END
    2B04 CD8225  SKD:CALL INTRAK        ;GET THE NEW PIXELS
30  2B07 CD8F26      CALL GETFTA        ;MAKE SURE WE GOT
                                        THERE
                     JRC SKFTA          ;JUMP IF ERROR
    2B0A+380D
    2B0C B7          ORA A              ;ALSO CHECK FOR TRACK 0
35                   JRNZ SKFTA         ;THIS IS ALSO ERROR
    2B0D+200A
    2B0F CD0F27      CALL BETBTA        ;WE NEED THIS ONE TOO
                     JRC SKFTA          ;JUMP IF PROBLEM
```

```
2B12+3805
2B14 B7          ORA A              ;OR IF NOT TRACK 0
                 JRNZ SKFTA
2B15+2002
2B17 AF          XRA A              ;ELSE SHOW GOOD FINISH
2B18 C9          RET
               SKFTA:DJNZ SKR1      ;JUMP IF NOT LAST TRY
2B19+100C
2B1B CDA528      CALL INTRK         ;IN TWO TRACKS
2B1E CDA528      CALL INTRK
2B21 110000      LXI D,0            ;AND DO IT AGAIN
2B24 C3C52A      JMP SK7
               SKR1:DJNZ SKR2       ;JUMP IF NOT PASS 2
2B27+1009
2B29 CD6B28      CALL OUTONE        ;GO OUT NET ONE
2B2C CD6B28      CALL OUTONE
2B2F 04          INR B              ;MAKE NEXT ONE LAST
                                    PASS
                 JMPR SKD
2B30+18D2
2B32 CD7128    SKR2:CALL INONE      ;STEP FOR 2ND PASS
2B35 04          INR B              ;SET UP FOR 2ND PASS
                                    FLAG
                 JMPR SKD
2B36+18CC
               SK9:NEG              ;GET POSITIVE SKEW
                                    DISTANCE
2B38+ED44
2B3A 47          MOV B,A            ;SAVE THE BYTE
2B3B AF3D        XRA A ! DCR A      ;SET CY BIT TO SHOW
                                    CCW ROTATION
2B3D 78          MOV A,B            ;RECOVER VALUE
```

```
1   CP/M MACRO ASSEM 2.0  #033  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
    2B3E CDD82B       CALL LOOKUP       ;SKEW STEPS TO A
    2B41 47           MOV B,A           ;COUNTER TO B
    2B42 CD6728  SK14:CALL SKEWOUT      ;ROTATE FILM CCW
5                     DJNZ SK14         ;UNTIL DONE
    2B45+10FB
    2B47 79      SK15:MOV A,C           ;GET THE CARRIAGE
                                        CORRECTION
    2B48 B7           ORA A             ;DONE?
10                    JRZ SKDONE        ;JUMP IF FINISHED
    2B49+28B7
    2B48 CD6B28       CALL OUTONE       ;ELSE STEP THE CARRIAGE
    2B4E 0D           DCR C             ;COUNT THE STEP
                      JMPR SK15         ;AND LOOP TILL DONE
15  2B4F+18F6
                  ;
                  ;
    2B51 110000  SKEW2:LXI D,0          ;CLEAR BOTH COUNTERS
    2B54 010000       LXI B,0
20  2B57 210000       LXI H,0           ;CLEAR VALID TRACK FLAG
    2B5A CD8225  GET1ST:CALL INTRAK     ;GET THE PIXELS
    2B5D 7C           MOV A,H           ;CHECK IF WE GOT FRONT
                                        ADDRESS
    2B5E B7           ORA A
25                    JRNZ GETBAK       ;JUMP IF WE GOT A VALID
                                        ADDRESS
    2B5F+2007
    2B61 CD8F26       CALL GETFTA       ;ELSE TRY FOR IT
                      JRC GETBAK        ;JUMP IF NOT GOOD
30                                      ADDRESS
    2B64+3802
    2B66 47           MOV B,A           ;ELSE STORE IT IN B
                                        REGISTER
    2B67 24           INR H             ;SHOW GOOD FTA
35  2B68 7D      GETBAK:MOV A,L         ;DO WE HAVE BTA YET?
    2B69 B7           ORA A
                      JRNZ BAK2         ;JUMP IF WE HAVE VALID
                                        BTA
```

```
1   2B6A+2007
    2B6C CD0F27      CALL GETBTA      ;TRY FOR VALID BACK
                                      TRACK ADDRESS
                     JRC BAK2         ;JUMP IF INVALID
                                      ADDRESS
5   2B6F+3802
    2B71 4F          MOV C,A          ;ELSE STORE GOOD
                                      ADDRESS IN C
    2B72 2C          INR L            ;SHOW GOOD BTA
    2B73 7C      BAK2:MOV A,H         ;CHECK IF FIRST IS GOOD
10  2B74 B7          ORA A
                     JRZ BAK3         ;JUMP IF NO VALID FTA
    2B75+2807
    2B77 7D          MOV A,L          ;IS BTA VALID?
    2B78 B7          ORA A
15                   JRNZ SKEWDONE    ;JUMP IF WE HAVE BOTH
    2B79+200D
    2B78 14          INR D            ;BUMP FTA OFFSET COUN-
                                      TER IF NO BTA
                     JMPR BAK4        ;STEP CARRIAGE AND TRY
20                                    AGAIN
    2B7C+1805
    2B7E 7C      BAK3:MOV A,H         ;CHECK FOR BTA GOOD
    2B7F B7          ORA A
                     JRZ BAK4         ;JUMP IF NO VALID BTA
25  2B80+2801
    2B82 1C          INR E            ;BUMP THE BTA OFFSET
                                      COUNTER
    2B83 CD6B28  BAK4:CALL OUTONE     ;MOVE CARRIAGE ONE STEP
                                      MORE
30                   JMPR GET1ST      ;TRY AGAIN IN NEW POSI-
                                      TION
    2B86+1802
                                      ;NEITHER ONE IS GOOD
    2B88 79      SKEWDONE:MOV A,C     ;BACK ADDRESS TO A
35  2B89 90          SUB B            ;FIND THE OFFSET: FTA
                                      TO BTA
                     JRNZ SKEWYOU     ;JUMP IF NON ZERO TRACK
                                      OFFSET
```

```
1   CP/M MACRO ASSEM 2.0  #034  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

    2B8A+200B
    2B8C 4F          MOV C,A            ;SAVE BYTE
    2B8D 7AB3        MOV A,D ! ORA E    ;ANY OFFSET AT ALL?
5                    JRNZ SKEWME        ;JUMP IF SOMETHING
    2B8F+2004
    2B91 4F          MOV C,A            ;SHOW NO CARRIAGE
                                        EFFECT
    2B92 60          MOV H,B            ;GET FTA TO H REGISTER
10  2B93 78          MOV A,B            ;AND TO A
    2B94 C9          RET
                  SKEWME:MOV A,C        ;RECOVER 0
    2B96 B7          ORA A              ;SHOW WHAT IT IS
    2B97 F5       SKEWYOU:PUSH PSW      ;SAVE THE POLARITY
15                                      STATUS
                     BIT 7,A            ;NEGATIVE NUMBER?
    2B98+CB7F
                     JRZ BAKCCW         ;JUMP POLARITY OK
    2B9A+2802
20                   NEG                ;TURN NUMBER POSITIVE
    2B9C+ED44
    2B9E 60       BAKCCW:MOV H,B        ;SAVE FTA IN H REGISTER
    2B9F 0604        MVI B,TRACK        ;STEPS/TRACK TO B
    2BAl 4F          MOV C,A            ;OFFSET TO C REGISTER
25  2BA2 3E00        MVI A,0            ;CLEAR RESULT REGISTER
    2BA4 81       MULTI:ADD C           ;ADD EM UP
                     DJNZ MULT1         ;UNTIL MULTIPLIER GONE
    2BA5+10FD
    2BA7 93          SUB E              ;ADD OFFSETS
30  2BA8 82          ADD D              ;TRACK ENDS ARE A STEPS
                                        APART
                                        ;ROTATED CW.(BTA CW
                                        FROM FTA)
    2BA9 CDD82B      CALL LOOKUP        ;TRANSLATE INTO SKEW
35                                      STEPS IN A,
                                        ;CARRIAGE OFFSET IN C
    2BAC FE6C        CPI SKLIMIT/2      ;TOO FAR?
                     JRC S0K1           ;JUMP IF OK
```

```
1   2BAE+3805
    2BB0 F1      TOOBIG:POP PSW        ;FIX STACK
    2BB1 AF3D        XRA A ! DCR A     ;SHOW BAD SKEW
    2BB3 37          STC
    2BB4 C9          RET
5   2BB5 47      SOK1:MOV B,A          ;MOVE SKEW COUNTER TO B
    2BB6 F1          POP PSW           ;RECOVER DIRECTION
                                       STATUS
                     JRC CCW2          ;JUMP IF ROTATION WAS
                                       CCW
10  2BB7+380E
    2BB9 CD6328  SKW:CALL SKEWIN       ;ROTATE CARRIAGE CW
                     DJNZ SKW          ;UNTIL PARALLEL
    2BBC+10FB
    2BBE 79          MOV A,C           ;GET CARRIAGE OFFSET
15                   EXX               ;WE WILL SHOW CW ROTA-
                                       TION AS
    2BBF+D9
    2BC0 47          MOV B,A           ;A NON-ZERO IN THE B
                                       REGISTER
20  2BC1 0E00        MVI C,0           ;WITH THE C REG.
                                       CLEARED
                     EXX
    2BC3+D9
    2BC4 AF          XRA A             ;CLEAR CY TO SHOW GOOD
25  2BC5 7C          MOV A,H           ;RECOVER THE FTA
    2BC6 C9          RET               ;RETURN WITH CARRIAGE
                                       CORRECTION
                                       ;IN C REGISTER,FTA IN
                                       A REGISTER
30  2BC7 CD6728  CCW2:CALL SKEWOUT     ;ROTATE CARRIAGE CCW
                     DJNZ CCW2         ;UNTIL PARALLEL WITH
                                       HEAD ROTATION
    2BCA+10FB
    2BCC 79          MOV A,C           ;GET THE CARRIAGE
35                                     POSITION
                     EXX               ;WE WILL SHOW CCW
                                       ROTATION
```

```
1   CP/M MACRO ASSEM 2.0  #035  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

    2BCD+D9
    2BCE 4F          MOV C,A          ;BY NON-ZERO IN THE
                                      C REGISTER
5   2BCF 0600        MVI B,0          ;CLEAR B REGISTER
                     EXX
    2BD1+D9
                     NEG              ;MAKE IT NEGATIVE
                                      OFFSET
10  2BD2+ED44
    2BD4 4F          MOV C,A          ;BACK TO C REGISTER
    2BD5 AF          XRA A            ;CLEAR CY FOR GOOD
                                      RETURN
    2BD6 7C          MOV A,H          ;RECOVER THE FTA TO A
15                                    REGISTER
    2BD7 C9          RET              ;RETURN AS PER ABOVE
                     ;
                     ; SUBROUTINE LOOKUP
                     ;
20                   ; SUBROUTINE LOOKUP CONVERTS CARRIAGE STEP
                     OFFSET (PASSED
                     ; IN THE A REGISTER) TO SKEW STEPS (IN THE
                     A REGISTER)
                     ; AND CARRIAGE OFFSET CORRECTION (IN THE C
25                   REGISTER).
                     ; THE CALLING ROUTINE SHOULD SET THE CY BIT
                     TO INDICATE
                     ; WHICH DIRECTION OF TRAVEL IS DESIRED. CY
                     BIT SET INDICATES
30                   ; CCW ROTATION, WHILE CY BIT RESET (NO CY)
                     INDICATES CW
                     ; ROTATION. IF UNSUCCESSFUL (IE: STEPPER
                     OVERTRAVEL),
                     ; THE ROUTINE WILL RETURN WITH THE CY SET
35                   AND NON-ZERO
                     ; IN THE A REGISTER.
                     ;
    2BD8 E5          LOOKUP:PUSH H    ;SAVE ALL
```

```
2BD9 D5          PUSH D
2BDA C5          PUSH B
                 IF HALFSTP
                 PUSH PSW          ;SAVE STATUS
                 RAR ! ANI 7FH     ;DIVIDE BY 2
                 MOV B,A           ;SAVE HERE
                 POP PSW           ;RECOVER STATUS
                 MOV A,B           ;NEW BYTE TO A
                 ENDIF
2BDB 2AB440      LHLD BITBYTE      ;GET BREAKPOINT BYTE
                                   & BIT COUNTER
2BDE EB          XCHG              ;COUNTER TO D, BYTE TO
                                   E
2BDF 2AB240      LHLD SKADDR       ;GET TABLE ADDRESS
                                   POINTER
                 JRC LOOKCCW       ;JUMP IF CCW ROTATION
2BE2+383E
2BE4 B7          ORA A             ;CORRECTION REQUIRED?
                 JRNZ LCW          ;JUMP IF CW CORRECTION
                                   REQUIRED
2BE5+2013
2BE7 57          LDONE:MOV D,A     ;# OF STEPS IN D
2BE8 C1          POP B             ;USERS B REGISTER RE-
                                   STORED
2BE9 0E00        MVI C,0           ;CLEAR RESULT REGISTER
2BEB D609        SUI 9             ;GET FIRST HALF STEP OF
                                   EFFECT
                 JMPR LOOK0        ;TEST IT & ENTER LOOP
2BED+1802
                 IF NOT HALFSTP
2BEF D613        LOOK4:SUI 19      ;SUBTRACT ONE STEP OF
                                   CARRIAGE EFFECT
                 ELSE
                 LOOK4:SUI 10
                 ENDIF
                 LOOK0:JRC LOOK5   ;JUMP IF WE ARE DONE
2BF1+3803
2BF3             INR C             ;ADD THE CARRIAGE STEP
                 JMPR LOOK4        ;LOOP TILL ALL FOUND
```

```
1    2BF4+18F9
     2BF6 7A      LOOK5:MOV A,D        ;RECOVER # OF SKEW
                                       STEPS TO A
     2BF7 D1          POP D            ;RECOVER OTHER USER
                                       REGS
```

```
1   CP/M MACRO ASSEM 2.0  #036  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI

    2BF8 E1          POP H
    2BF9 C9          RET
    2BFA 47      LCW:MOV B,A            ;CARRIAGE STEPS TO B
5               LCW1:RLCR E             ;BREAKPOINT BIT TO CY
    2BFB+CB03
    2BFD CE03        ACI 3              ;PLUS STEP CONSTANT
                     DJNZ LOOK3         ;JUMP IF CARRIAGE STEPS
                                        NOT DONE
10  2BFF+100A
    2C01 22B240  LSAVE:SHLD SKADDR      ;SAVE THE TABLE POINTER
    2C04 626B        MOV H,D ! MOV L,E  ;GET THE BREAKPOINT
                                        BYTE & COUNTER
    2C06 22B440      SHLD BITBYTE       ;AND SAVE THEM TOO
15                   JMPR LDONE         ;FINISH UP & RETURN
    2C09+18DC
    2C0B 15      LOOK3:DCR D            ;CHECK IF BREAKPOINT
                                        BITS DONE
                     JRNZ CCW1          ;JUMP IF BYTE STILL OK
20  2C0C+2015
    2C0E 1608        MVI D,8            ;RESET BIT COUNTER
    2C10 2B          DCX H              ;TO NEXT TABLE BYTE
    2C11 5E          MOV E,M            ;GET IT
                     EXAF               ;SAVE THE SKEW COUNT
25  2C12+08
    2C13 3E13        MVI A,(SKTABLE-5)AND OFFH ;END OF TABLE
                                        ADDR.
    2C15 BD          CMP L              ;WATCH TABLE POINTER
                     JRZ LERROR         ;JUMP IF OUT OF TABLE
30                                      RANGE
    2C16+2803
                     EXAF               ;RECOVER SKEW COUNT
    2C18+08
                     JMPR LCW1          ;AND CONTINUE
35  2C19+18E0
    2C1B C1D1E1  LERROR:POP B ! POP D ! POP H ;RECOVER ALL
    2C1E AF          XRA A              ;GET 0 TO CARRIAGE
                                        CORRECTION
```

```
2C1F 4F          MOV C,A
2C20 3D          DCR A              ;SHOW UNSUCCESSFUL TRY
2C21 C9          RET
2C22 47      LOOKCCW:MOV B,A        ;CARRIAGE STEPS TO B
             CCW1:RRCR E            ;BREAKPOINT BIT TO CY
2C23+CB0B
2C25 CE03        ACI 3              ;PLUS CONSTANT
                 DJNZ LOOK8         ;LOOP TILL ALL STEPS
                                    DONE
2C27+1002
                 JMPR LSAVE         ;THEN FINISH UP
2C29+18D6
2C2B 14      LOOK8:INR D            ;CHECK BREAKPOINT BITS
                 BIT 3,D            ;DONE YET?
2C2C+CB5A
                 JRZ CCW1           ;LOOP IF BYTE STILL OK
2C2E+28F3
                 BIT 0,D            ;CHECKING FOR A 9
2C30+CB42
                 JRZ CCW1
2C32+28EF
2C34 1601        MVI D,1            ;ELSE RESET COUNTER
2C36 23          INX H              ;AND GET NEXT
2C37 5E          MOV E,M
                 EXAF               ;SAVE THE STEPPER
                                    COUNT
2C38+08
2C39 3E1D        MVI A,(DKTABLE+5) AND OFFH ;TOO FAR?
2C3B BD          CMP L
                 JRZ LERROR         ;JUMP IF TOO FAR
2C3C+28DD
```

```
1   CP/M MACRO ASSEM 2.0  #037  READ11 READER SOFTWARE
                                10/23/80 AAJ (C) 1980 NLI
                    EXAF              ;RECOVER COUNT IF OK
    2C3E+08
                    JMPR CCW1         ;AND LOOP TILL DONE
5   2C3F+18E2
                ;
                ;
                ; SUBROUTINE CEFFECT
                ;
10              ; SUBROUTINE CEFFECT MOVES THE CARRIAGE TO
                CORRECT
                ; SKEW MOTOR INTERACTION WITH EFFECTIVE
                CARRIAGE POSITION.
                ;
15              CEFFECT:EXX       ;GET THE ALTERNATE
                                  REGISTERS
    2C41+D9
    2C42 78         MOV A,B       ;CHECK FOR CW ROTATION
                                  (NEEDS IN)
20  2C43 B7         ORA A
                    JRZ NOTCW     ;JUMP IF NOT CLOCKWISE
                                  ROTATION
    2C44+280A
    2C46 79     CWLOOP:MOV A,C    ;CHECK FOR DONE
25  2C47 B7         ORA A
                    JRZ CEFDONE   ;JUMP IF SO
    2C48+2810
    2C4A 0D         DCR C         ;ELSE COUNT STEP
    2C4B CD7128     CALL INONE    ;STEP THE CARRIAGE
30                  JMPR CWLOOP   ;TILL DONE
    2C4E+18F6
    2C50 78     NOTCW:MOV A,B     ;CHECK FOR DONE
    2C51 B7         ORA A
                    JRZ CEFDONE   ;JUMP IF SO
35  2C52+2806
    2C54 05         DCR B         ;COUNT THE STEP
    2C55 CD6B28     CALL OUTONE   ;MOVE THE CARRIAGE
                    JMPR NOTCW    ;LOOP TILL DONE
```

```
1   2C58+18F6
                    CEFDONE:EXX             ;RESTORE NORMAL
                                            REGISTERS
    2C5A+D9
    2C5B C9             RET
5                   ;
                    ;
                    ; SUBROUTINE CONPRNT
                    ;
                    ; THIS ROUTINE PRINTS A STRING TO THE CONSOLE.
                    THE
10                  ; STRING IS POINTED TO BY THE (DE) PAIR, AND
                    IS TERMINATED
                    ; WITH A 0 (NUL).
                    ;
15  2C5C 1A         CONPRNT:LDAX D          ;GET THIS CHARACTER
    2C5D 13             INX D               ;POINT TO NEXT
    2C5E B7             ORA A               ;CHECK FOR TERMINATOR
    2C5F C8             RZ                  ;RETURN IF DONE
    2C60 CD652C         CALL CHROUT         ;SEND CHARACTER OUT
20                      JMPR CONPRNT        ;LOOP TILL DONE
    2C63+18F7
                    ;
    2C65 4F         CHROUT:MOV C,A          ;CHARACTER TO PRINT TO
                                            C
25  2C66 C3692C         JMP CRT             ;TO THE CRT
                    ;
                    ;
                    ; THIS MODULE IS A CRT DRIVER FOR THE MICRO-
                    LINK 97098
30                  ; CRT CONTROLLER CARD. AS SUPPLIED, IT WILL
                    DRIVE THE
                    ; CARD AS SUPPLIED. IF YOU CHANGE THE MEMORY
                    ADDRESS
                    ; JUMPER, IT WILL REQUIRE CHANGING THE
35                  BASEAD EQUATE
```

```
1   CP/M MACRO ASSEM 2.0   #038  READ11 READER SOFTWARE
                                  10/23/80 AAJ (C) 1980 NLI
                ; TO REFLECT THE NEW BASE ADDRESS OF THE CARD.
                ;
                ; UPON POWER-UP, THE FIRST CHARACTER SENT
5                 TO THIS ROUTINE
                ; SHOULD BE A SCREEN CLEAR (1AH).  THIS IS
                  TO INIT THE
                ; VARIOUS POINTERS AND ALSO SET UP THE
                  MC6845 CRT
10              ; CONTROLLER CHIP.
                ;
                ;
    E000 =      BASEAD  EQU 0E000H      ;BOARD IS SHIPPED
                                         FOR 0A000H
15                                       ;CHANGE THIS EQUATE
                                         IF YOU CHANGE
                                         ;THE BOARD ADDRESS
                                         JUMPER
                ;
20  E480 =      SCRN    EQU BASEAD+480H;SCREEN LOCATION
    E000 =      ADDREG  EQU BASEAD      ;MC6845 ADDRESS REGI-
                                         STER.
    E001 =      PARAM   EQU BASEAD+1    ;PARAMETERS TO HERE
    000E =      HCURSOR EQU 14          ;THIS IS HI BYTE
25                                       OF CURSOR FLAG
    000F =      LCURSOR EQU 15          ;THIS IS LO BYTE FLAG
                ;
    E400 =      FREERAM EQU BASEAD+400H;WE USE SPARE CRT
                                         RAM FOR SCRATCH
30  E400 =      CFLAG   EQU FREERAM     ;CURSOR ADDRESS MODE
                                         FLAG
    E401 =      VPOS    EQU FREERAM+1   ;CURSOR VERTICAL
                                         POSITION
```

```
1    E402 =      HPOS    EQU FREERAM+2  ;CURSOR HORIZONTAL
                                         POSITION
     E403 =      HLSAVE  EQU FREERAM+3  ;SAVE USERS H&L HERE
     E410 =      LOCSTK  EQU FREERAM+16 ;LOCAL STACK HERE TOO
                 ;
5    001E =      CURHOME EQU 1EH        ;CURSOR HOME CHARACTER
     0015 =      NAK     EQU 15H        ;CLEAR TO END OF LINE
                                         CHARACTER
     0020 =      SPACE   EQU 20H        ; SPACE CHARACTER
     0008 =      BS      EQU 08H        ;BACK SPACE
10   001B =      ESC     EQU 1BH        ;ESCAPE CHARACTER FOR
                                         CURSOR ADD.
     000B =      VT      EQU 0BH        ;VERTICAL TAB (UP 1
                                         LINE)
     0009 =      HT      EQU 09H        ;HORIZONTAL TAB
15                                       (EVERY 8 COLUMNS)
     001A =      CLR     EQU 'Z'-40H    ;CONT-Z CLEARS SCREEN
                 ;
                 ;
                 ; THIS ROUTINE WILL EMULATE
20               ; AN ADM3A TERMINAL OR AN IMSAI VIO VIDEO
                   DISPLAY, INCLUDING
                 ; CURSOR ADDRESS MODE AND ERASE-TO-END-OF-LINE.
                 ;
     2C69 2203E4 CRT:    SHOULD HLSAVE  ;PUT AWAY USERS H&L
25   2C6C 210000         LXI H,P        ;CLEAR FOR REGISTER ADD
     2C6F 39             DAD SP         ;GET USERS SP
     2C70 3110E4         LSI SP,LOCSTK  ;SET NEW STACK
     2C73 79             MOV A,C        ;MOVE CHARACTER TO A
     2C74 E5             PUSH H         ;SAVE ALL REGS
30   2C75 D5             PUSH D
     2C76 C5             PUSH B
     2C77 F5             PUSH PSW
     2C78 3A00Ef         LDA CFLAG      ;CHECK FOR CURSOR
```

```
                                        ADDRESS MODE
        2C7B B7         ORA A
        2C7C C204A2D    JNC PRINT       ;JUMP IF SO
        2C7F 79         MOV A,C         ;RECOVER CHARACTER
        2C80 FE20       CPI SPACE       ;IS IT PRINTABLE
        2C82 D2582D     JNC PRINT       ;JUMP IF SO
        2C85 FE15       CPI NAK         ;CLEAR TO END OF LINE?
        2C87 CAAE2D     JZ LERASE       ;JUMP IF SO
        2C8A FE1E       CPI CURHOME     ;HOME CURSOR?
        2C8C CAF72C     JZ CHOME        ;JUMP IF SO
        2C8F FE09       CPI HT          ;HORIZONTAL TAB?
        2C91 CA9C2D     JZ HTAB         ;JUMP IF SO
```

```
1    CP/M MACRO ASSEM 2.0  #039  READ11 READER SOFTWARE
                                 10/23/80 AAJ (C) 1980 NLI
     2C94 FE0B           CPI VT          ;VERTICAL TAB?
     2C96 CA782D         JZ VTAB         ;
     2C99 FE0D           CPI CR          ;CARRIAGE RETURN
5                        JRZ CRET        ;JUMP IF SO
     2C9B+2819
     2C9D FE0A           CPI LF          ;LINE FEED?
                         JRZ LFEED       ;JUMP IF SO
     2C9F+281C
10   2CA1 FE08           CPI BS          ;BACK SPACE?
                         JRZ BACKSP      ;BACK UP IF SO
     2CA3+2828
     2CA5 FE1B           CPI ESC         ;CURSOR ADDRESS MODE?
                         JRZ CADD1       ;JUMP IF SO
15   2CA7+285A
     2CA9 D61A           SUI CLR         ;SCREEN CLEAR?
                         JRZ CLEAR       ;CLR SCREEN & RESET
     2CAB+282D
     2CAD F1     RETURN: POP PSW         ;RESTORE ALL REGISTERS
20   2CAE C1             POP B
     2CAF D1             POP D
     2CB0 E1             POP H
     2CB1 F9             SPHL            ;RETURN USERS SP
     2CB2 2A03E4         LHLD HLSAVE     ;AND HL
25   2CB5 C9             RET
                 ;
                 ;
                 ;
                 ;
30   2CB6 AF     CRET:   XRA A           ;GET 0 COLUMN
     2CB7 3202E4         STA HPOS        ;TO MEMORY
     2CBA C3D62D         JMP FINISH      ;FINISH UP
                 ;
```

```
                    ;
2CBD 2101E4   LFEED:  LXI H,VPOS      ;POINT TO ROW 3
2CC0 7#               MOV A,M         ;GET IT
2CC1 34               INR M           ;NEXT ROW
2CC2 FE17             CPI 23          ;LAST ROW?
                      JRNZ L1         ;JUMP IF NOT LAST ROW
2CC4+2004
2CC6 35               DCR M           ;FIX FOR LAST ROW
2CC7 CDED2D           CALL SCROLL     ;PUSH UP SCREEN
2CCA C3D62D L1:       JMP FINISH      ;NEW ADDR. TO CRT CHIP
                    ;
                    ;
2CCD 3A02E4   BACKSP; LDA HPOS        ;GET COLUMN POSITION
2CD0 B7               ORA A           ;CHECK FOR COLUMN 0
                      JRZ RETUPN      ;CAN'T BACKSPACE PAST
                                       MARGIN
2CD1+28DA
2CD3 3D               DCR A           ;BACK UP ONE POSITION
2CD4 3202E4           STA HPOS        ;PUT IN MEMORY
2CD7 C3D62D           JMP FINISH      ;TELL THE CRT CHIP
                                       ABOUT IT
                    ;
                    ;
2CDA 2180E4   CLEAR:  LXI H,SCRN      ;CLEAR THE SCREEN
2CDD 1181E4           LXI D,SCRN+1
2CE0 017F07           LXI B,(24*80)-1
2CE3 77               MOV M,A         ;0 TO MEMORY
                      LDIR
2CE4+EDB0
2CE6 0610             MVI B,16        ;WE MUST INIT. 16
                                       REGISTERS
```

```
1    CP/M MACRO ASSEM 2.0  #040 READ 11 READER SOFTWARE
                                  10/23/80 AAJ (C) 1980 NLI
     2CE8 21032E          LXI H,INITBL   ;POINT TO INIT.TABLE
     2CEB 79      INIT:   MOV A,C        ;GET THE COUNTER
     2CEC 0C              INR C          ;BUMP IT
5    2CED 3200E0          STA ADDREG     ;TELL THE CHIP WHICH
                                         REGISTER
     2CF0 7E              MOV A,M        ;GET THE TABLE VALUE
     2CF1 3201E0          STA PARAM      ;TO THE CHIP
     2CF4 23              INX H          ;POINT TO NEXT TABLE
10                                       VALUE
                          DJNZ INIT      ;LOOP TILL ALL DONE
     2CF5+10F4
     2CF7 AF      CHOME:  XRA A          ;GET 0 AGAIN
     2CF8 2100E4          LXI H,CFLAG    ;POINT TO CURSOR
15                                       ADDR. FLAG
     2CFB 77              MOV M,A        ;RESET IT
     2CFC 23              INX H          ;POINT TO VERTICAL
                                         POSITION
     2CFD 77              MOV M,A        ;RESET IT
20   2CFE 23              INX H          ;THEN HORIZ POSITION
     2CFF 77              MOV M,A
     2D00 C3D62D          JMP FINISH
                  ;
     2D03 3E01    CADD1:  MVI A,1        ;SET CURSOR ADDRESS
25                                       MODE
     2D05 3200E4          STA CFLAG
                          JMPR RETURN
     2D08+18A3
                  ;
30   2D0A 3A00E4 AMODE:   LDA CFLAG      ;GET FUNCTION #
     2D0D 3D              DCR A          ;CHECK FOR 1
                          JRNZ M1        ;JUMP IF NOT
     2D0E+2019

35
```

```
1    2D10 79              MOV A,C         ;RECOVER CHARACTER
     2D11 FE3D            CPI '='         ;CURSOR SEQUENCE?
                          JRNZ M2         ;PRINT IF NOT
     2D13+2007
     2D15 3E02            MVI A,2         ;UPDATE MEMORY
5    2D17 3200E4          STA CFLAG
                          JMPR RETURN
     2D1A+1891
     2D1C AF      M2:     XRA A           ;GET 0
     2D1D 3200E4          STA CFLAG       ;RESET POINTER
10   2D20 79              MOV A,C         ;RECOVER CHARACTER
                          JMPR PRINT      ;PRINT IT
     2D21+1835
     2D23 AF      M9:     XRA A           ;GET 0
     2D24 3200E4          STA CFLAG       ;RESET FLAG
15                        JMPR RETURN     ;DONE
     2D27+1884
     2D29 3D      M1:     DCR A           ;CHECK FOR 2
                          JRNZ M3         ;JUMP IF NOT
     2D2A+2011
20   2D2C 79              MOV A,C         ;GET VERTICAL VALUE
     2D2D D620            SUI SPACE       ;REMOVE BIAS
                          JRC M9          ;ABORT IF ERROR
     2D2F+38F2
     2D31 FE18            CPI 24          ;CHECK IF TOO BIG
25                        JRNC M9         ;JUMP IF TOO BIG
     2D33+30EE            SETB 7,A        ;FLAG HI BIT OF THIS VALUE
     2D35+CBFF
     2D37 3200E4          STA CFLAG       ;SAVE FOR NEXT
     2D3A C3AD2C          JMP RETURN
30   2D3D 3A00E4  M3:     LDA CFLAG       ;GET OLD VALUE
     2D40 6F              MOV L,A         ;TO L REGISTER
```

```
1    CP/M MACRO ASSEM 2.0  #041    READ 11 READER SOFTWARE
                                   10/23/80 AAJ (C) 1980 NLI
                         RES 7,L        ;KILL BIAS
     2D41+CBBD
     2D43 79             MOV A,C        ;GET NEW BYTE
5    2D44 D620           SUI SPACE      ;REMOVE BIAS
                         JRNC M4        ;JUMP IF NOT TOO SMALL
     2D46+3002
                         JMPR M9        ;RESET & EXIT IF SMALL
     2D48+18D9
10   2D4A FE50   M4:     CPI 80         ;TOO LARGE?
                         JRNC M9        ;JUMP IF SO
     2D4C+30D5
     2D4E 67             MOV H,A        ;SAVE IF OK
     2D4F AF             XRA A          ;GET 0
15   2D50 3200E4         STA CFLAG      ;RESET CURSOR ADDRESS
                                        FLAG
     2D53 2201E4         SHLD VPOS      ;UPDATE TO MEMORY
                         JMPR FINISH    ;FINISH UP
     2D56+187E
20                 ;
     2D58 CDBF2D PRINT:  CALL CURADD    ;GET CURRENT ADDRESS
     2D5B D620           SUI SPACE      ;FIX CHARACTER
                                        FOR CRT
     2D5D77              MOV M,A        ;PRINT CHARACTER
25   2D5E 2101E4         LXI H,VPOS     ;POINT TO POINTERS
     2D61 0C             INR C          ;NEXT COLUMN POS.
     2D62 79             MOV A,C        ;GET BYTE TO TEST
     2D63 FE50           CPI 80         ;LAST COLUMN?
                         JRNZ P1        ;JUMP IF OK
30   2D65+200D
     2D67 0E00           MVI C,0        ;CHANGE COLUMN 80 TO 0
     2D69 7E             MOV A,M        ;GET VERTICAL POINTER
     2D6A FE17           CPI 23         ;AT BOTTOM?

35
```

```
                        JRZ P2          ;SCROLL IF SO
    2D6C+2803
    2D6E 34             INR M           ;BUMP IT
                        JMPR P1
    2D6F+1803
    2D71 CDED2D P2:     CALL SCROLL     ;MOVE IT UP
    2D74 23     P1:     INX H           ;POINT TO COLUMN LOCA-
                                         TION
    2D75 71             MOV M,C         ;NEW COLUMN #
                        JMPR FINISH     ;DONE
    2D76+185E
                ;
                ;
    2D78 2101E4 VTAB:   LXI H,VPOS      ;POINT TO VERTICAL
                                         POSITION
    2D7B 7E             MOV A,M         ;GET POINTER
    2D7C B7             ORA A           ;CHECK FOR TOP OF
                                         SCREEN
                        JRNZ V1         ;JUMP IF NOT TOP
    2D7D+201A
    2D7F 21AFEB         LXI H,SCRN+(23*80)-1    ; MOVE IT
                                                  DOWN
    2D82 11FFEB         LXI D,SCRN+(24*80)-1
    2D85 013007         LXI B,23*80
                        LDDR
    2D88+EDB8
    2D8A 2180E4         LXI H,SCRN      ;POINT TO FIRST LINE
    2D8D 1181E4         LXI D,SCRN+1
    2D90 014F00         LXI B,79
    2D93 3600           MVI M,0         ;CLEAR FIRST
                        LDIR            ;BLOCK MOVE DOES REST
    2D95+EDB0
                        JMPR FINISH     ;DONE
    2D97+183D
```

```
1    CP/M MACRO ASSEM 2.0    #042    READ 11 READER SOFTWARE
                                     10/23/80 AAJ (C) 1980 NLI
     2D99 35      V1:      DCR M          ;ONE ROW LESS
                           JMPR FINISH
     2D9A+183A
5                 ;
     2D9C 2102E4  HTAB:    LXI H,HPOS     ;GET HORIZONTAL
                                           POSITION
     2D9F 34      HTAB1:   INR M          ;BUMP IT
     2DA0 7E               MOV A,M        ;CHECK FOR TAB STOP
10   2DA1 FE50             CPI 80         ;TOO FAR?
                           JRNZ HTAB2     ;JUMP IF OK
     2DA3+2003
     2DA5 35               DCR M          ;FIX IT
                           JMPR FINISH
15   DA6+182E
     2DA8 E607    HTAB2:   ANI 7          ;ARE WE OK?
                           JRNZ HTAB1     ;LOOP TILL FOUND
     2DAA+20F3
                           JMPR FINISH    ;SET IT & EXIT
20   2DAC+1828

                  ;
     2DAE CDBF2D  LERASE:  CALL CURADD    ;GET ADDRESS
     2DB1 79               MOV A,C        ;THIS COLUMN #
25   2DB2 D650             SUI 80         ;SUBTRACT BIAS
                           NEG            ;MAKE INTO POSITIVE
                                           NUMBER
     2DB4+ED44
     2D86 47               MOV B,A        ;SET UP AS COUNTER
30   2DB7 3600    LOOP:    MVI M,0        ;CLEAR POSITION
     2DB9 23               INX H          ;POINT TO NEXT
                           DJNZ LOOP      ;DO TILL END OF LINE
```

```
2DBA+10FB
2DBC C3AD2C          JMP RETURN
                ;
                ;CURADD RETURNS THE CURRENT CURSOR ADDRESS
                 IN (HL)
                :
2DBF 2A01E4 CURADD: LHLD VPOS       ;GET HORIZ & VERT
                                     FLAGS
2DC2 4C             MOV C,H         ;SAVE HORIZ IN C
2DC3 2600           MVI H,0         ;CLEAR HI BYTE
2DC5 1180E4         LXI D,SCRN      ;SCREEN LOCATION IN DE
2DC8 29             DAD H           ;*16
2DC9 29             DAD H
2DCA 29             DAD H
2DCB 29             DAD H
2DCC EB             XCHG            ;SWAP DE / HL
2DCD 19             DAD D           ;ADD * 16 TO BASE ADD.
2DCE EB             XCHG            ;SWAP BACK
2DCF 29             DAD H           ;MAKE * 64
2DD0 29             DAD H
2DD1 19             DAD D           ;ADD TO MAKE * 80
2DD2 0600           MVI B,0         ;CLEAR HI BYTE
2DD4 09             DAD B           ;ADD COLUMN POSITION
2DD5 C9             RET
                ;
                ; FINISH SETS THE CURSOR ADDRESS TO THE
                  CURRENT
                ; RAM VALUES, THEN EXITS TO RETURN TO THE
                  USER.
                ;
2DD6 CDBF2D FINISH: CALL CURADD     ;GET THE CURRENT
                                     ADDRESS IN (HL)
2DD9 11001C         LXI D,-(SCRN-80H) ; REMOVE BIAS
2DDC 19             DAD D
```

```
1    2DDD 1100E0        LXI D,ADDREG    ;THIS IS CRT CONTROL-
                                        LER PORT
     2DE0 EB            XCHG            ;CRT CHIP AT (HL),
                                        CURSOR ADD IN DE
```

```
1    CP/M MACRO ASSEM 2.0  #043  READ 11 READER SOFTWARE
                                 10/23/80 AAJ (C) 1980 NLI
     2DE1 360F          MVI M,LCURSOR  ;WE SEND LO BYTE FIRST
     2DE3 23            INX H          POINT TO PARAMETER PORT
     2DE4 73            MOV M,E        ;LO BYTE TO CHIP
5    2DE5 2B            DCX H          ;BACK TO ADDRESS PORT
     2DE6 360E          MVI M,HCURSOR  ;WE SEND HI BYTE NOW
     2DE8 23            INX H          ;BACK TO PARAMETER PORT
     2DE9 72            MOV M,D        ;HI BYTE TO CHIP
     2DEA C3AD2C        JMP RETURN     ;THEN EXIT
10                ;
                  ;
     2DED 1180E4 SCROLL: LXI D,SCRN    ;TOP OF SCREEN
     2DF0 21D0E4         LXI H,SCRN+80 ;2ND LINE
     2DF3 013007         LXI B,23*80
15                       LDIR          ;MOVE IT
     2DF6+EDB0
     2DF8 62             MOV H,D       ;LOC OF LAST LINE
     2DF9 6B             MOV L,E       ;TO HL
     2DFA 13             INX D         ;ONE MORE TO DE
20   2DFB 014F00         LXI B,79      ;ONE LINE LONG
     2DFE 3600           MVI M,0       ;CLEAR FIRST CHAR
                         LDIR          ;BLOCK MOVE DOES REST
     2E00+EDB0
     2E02 C9             RET
25                ;
                  ; THIS IS THE INIT TABLE FOR THE CRT
                    CONTROLLER.
                  ; THE VALUES REPRESENT THE 16 REGISTERS
                    WHICH MUST
30                ;BE SET UP FOR THE CRT CHIP TO FUNCTION.
                  ;
                  ;       VALUE PORT #     FUNCTION
                  ;
```

```
1     E03 6B      INITBL: DB 80+27 ; 0   ;HORIZONTAL TOTAL
      2E04 50             DB 80    ; 1   ;HORIZONTAL DISPLAYED
                                           CHARACTERS
      2E05 55             DB 85    ; 2   ;HORIZONTAL SYNC
                                           POSITION
5     2E06 08             DB 08    ; 3   ;HORIZONTAL SYNC
                                           WIDTH
      2E07 1A             DB 26    ; 4   ;VERTICAL TOTAL
      2E08 12             DB 12H   ; 5   ;VERTICAL SCAN
                                           LINE ADJUST
10    2E09 18             DB 24    ; 6   ;VERTICAL DISPLAYED
                                           LINES
      2E0A 19             DB 25    ; 7   ;VERTICAL SYNC
                                           POSITION
      2E0B 01             DB 1     ; 8   ;INTERLACE MODE FLAG
15    2E0C 0B             DB 0BH   ; 9   ;MAX  SCAN LINE
                                           ADDRESS
      2E0D 00             DB 0     ; 10  ;CURSOR START LINE
      2E0F 0B             DB 0,80H ; 11  ;CURSOR END LINE
      2E0F 0080           DB 0,80H ;12,13;THIS  IS THE START-
20                                          ING RAM ADDRESS
      2E11 0080           DB 0,80H ;14,15;THIS IS CURSOR
                                           POSITION

                  ;
                  ;
25                PAGE
```

```
1     CP/M MACRO ASSEM 2.0    #044  READ 11 READER SOFTWARE
                              10/23/80 AAJ (C) 1980 NLI
            ::::::::::::::::::::::::::::::::::::::::
            :::                                    ::
            :::          RAM BUFFERS               ::
5           :::                                    ::
            ::::::::::::::::::::::::::::::::::::::::
            :
            :
      2E13 24              DB 00100100B    ;SKEW ADJUST
10    2E14 A5              DB 10100101B    ;BREAKPOINTS
      2E15 2A              DB 00101010B
      2E16 AA              DB 10101010B
      2E17 AB              DB 10101011B
      2E18 56     SKTABLE: DB 01010110B
15    2E19 B6              DB 10110110B
      2E1A DB              DB 11011011B
      2E1B B7              DB 10110111B
      2E1C 7B              DB 01111011B
                  ;
20    2E1D 00     ROMEND:  DB 0            :THIS IS END OF ROM
                                              AREA
                  ;
      4000        ORG      RAM
                  ;
25    4000                 DS 64           ;THIS IS STACK SPACE
                  STACK:
                  ;
      4040        OUTBUF:  DS LLENGTH+4    ;THIS IS OUTPUT BUFFER
                  ;
30    40A4        TRYNO:   DS 1            ;RETRY FLAG
                  ;
      40A5        HBYTES:  DS 1            ;HAMMING MUX COUNT
                  ;
      40A6        LEN:     DS 1            ;BIT LENGTH COUNTER
35                ;
```

```
                40A7    FUSE:   DS 1            ;THIS IS SAFETY VALVE
                        ;
                40A8    ERRFLAG:DS 1            ;THIS IS ERROR FLAG
                                                 FOR RETRIEVES
                        ;
                40A9    ERCOUNT:DS 1            ;THIS IS HAMMING
                                                 SCORE COUNTER
                        :
                40AA    POINTER:DS 2            ;LEADER/TAILER PIXEL
                                                 POINTER
                        ;
                40AC    LSTART: DS 2            ;LINE START ADDRESS
                        ;
                40AE    SAVHL:  DS 2            ;BIT POSITION STORAGE
                        ;
                40B0    CURLINE:DS2             ;CURRENT TRACK NUMBER
                        ;
                40B2    SKADDR: DS2             ;SKEW TABLE ADDRESS
                                                 POINTER STORAGE
                        ;
                40B4    BITBYTE:DS 2            ;BREAKPOINT BYTE &
                                                 BIT COUNTER
                        ;
                40B6    HMATRIX:DS 16           ;HAMMING CODE WORK-
                                                 SPACE
                        ;
                4100    ORG     ($ AND OFF00H) + 100 H ; TO PAGE
                                                         BORDER
                        ;
                4100    INBUF: DS INBLEN        ;12K BUFFER
                        ;
                6500    END     START
```

CLAIMS:

1. Apparatus for reading a data record on which the data is inscribed in an arcuate pattern on a data carrier, comprising: a data carrier support for receiving and holding a data carrier; a light source operable to emit a beam of light; beam directing means for receiving and directing the beam of light toward the supported data carrier, including sweep means for moving the light beam across the data carrier in arcs corresponding to the arcuate pattern; and receiving means mounted to receive the light beam after the same strikes the data carrier.

2. The apparatus of claim 1, wherein: said light source is fixed; and said beam sweep means is rotatably mounted.

3. The apparatus of claim 2, wherein: said light source is fixed to said rotatably mounted beam sweep means.

4. The apparatus of claim 1, and further comprising: a carriage mounting said data carrier support for movement radially of the arcuate pattern.

5. Apparatus for reading digital data which is inscribed in an arcuate pattern on a data carrier, comprising; a source of light operable to emit a beam of light; mounting means mounting said light source and the data carrier for movement one relative to the other such that the beam of light impinges on the data carrier in accordance with the arcuate pattern and the data correspondingly modulates the beam of light; and light receiving means mounted to receive the modulated light beam and convert the same into electrical signals corresponding to the data carried by the light beam.

6. The apparatus of claim 5, wherein said light

source comprises a laser.

7. The apparatus of claim 5, wherein said mounting means comprises: a frame; a carriage for holding the data carrier, said carriage mounted on said frame for movement radially of the arcuate pattern; and light beam directing means mounted on said frame for rotation to rotate the light beam for arcuate scanning of the data carrier.

8. The apparatus of claim 7, wherein: said carriage includes a pivotally mounted data carrier support; first motor means connected to said carriage for moving the same in the radial direction of the arcuate pattern; and second motor means connected to said support for pivoting the same to correct for skew.

9. The apparatus of claim 7, wherein said light beam directing means comprises: a substantially disc-shaped member carrying said light source, said disc-shaped member rotatably mounted on said frame.

10. The apparatus of claim 7, wherein said light beam directing means, comprises: a member mounted on said frame for rotation; and an optical system for receiving the light beam and directing the same along a path toward and which follows the arcuate pattern of the data carrier.

11. The apparatus of claim 10, wherein said light source is mounted on said member.

12. The apparatus of claim 10, wherein: said light source is fixed on said frame; and said optical system includes the axis of rotation of said member.

13. The apparatus of claim 10, wherein: said carriage includes a reflective surface for supporting the data carrier and for reflecting the beam of light after passage through the data carrier; and said light

receiving means includes a light receiver mounted on said member.

14. The apparatus of claim 10, wherein said light beam directing means comprises a light directing member in said optical system mounted on said rotatable member for movement in the radial direction with respect to the axis of rotation; and motor means connected to said light directing member and operable to move the same to compensate for manufacturing tolerances of the data carrier and for size changes due to ambient temperature.

15. The apparatus of claim 14, wherein said light directing means comprises a beam directing member and a pivotally mounted member carrying said beam directing member, and wherein said motor means comprises an output shaft coupled to pivot said pivotally mounted member.

16. The apparatus of claim 15, wherein said beam directing member comprises: a prism.

17. Apparatus for optically reading a data record having data inscribed on a data carrier in an arcuate pattern of parallel spaced-apart arcuate data paths, comprising: a frame structure including a base plate and a rotatably mounted member mounted for rotation parallel to said base plate, and drive means for rotating said member; light beam generating means mounted on said frame structure and operable to emit a light beam; a carriage to carry the data record mounted on said frame structure for movement radially of the arcuate pattern of the data; carriage drive means mounted on said frame structure and connected to and operable to move said carriage in increments each equal to at least an equal division of the incremental distance between the data

paths; beam directing means mounted on said rotatably mounted member to rotate therewith for directing the light beam along arcuate paths, including beam adjustment means; and light beam receiving means mounted on said frame structure for receiving the light beam after the same strikes and is modulated by the data record.

18. The apparatus of claim 17, and further comprising: a magnifier adapter mounted on and extending through said base plate for releasably receiving a magnifier as an aid in adjusting said beam adjustment means for adjusting the light beam to the focal plane of said data record.

19. The apparatus of claim 17, wherein said carriage drive means comprises: a stepping motor mounted on said base plate and connected to said carriage.

20. The apparatus of claim 17, wherein said carriage includes a pivotally mounted data record support; and further comprising: skew correction means connected to said data record support and operable to align the data record such that the arcuate path of the liquid beam has the same axis of rotation as the arcuate data paths of the data record.

21. The apparatus of claim 20, wherein said skew correction means comprises: a motor mounted on said carriage and coupled to said data record support.

22. The apparatus of claim 21, wherein said skew correction means further comprises: lever means pivotally mounted on said carriage and contacting said data record support; and an output shaft of said motor for moving said lever means.

23. The apparatus of claim 22, wherein said skew correction means further comprises: bias means

urging said data record carrier towards said motor.

24. The apparatus of claim 17, wherein said carriage comprises: guide track means mounted on said base plate; and wheel means rollingly supporting said carriage along said guide track.

25. The apparatus of claim 24, wherein said wheel means comprises: a plurality of guide wheels each engaging said guide track means; and a plurality of respective adjustable eccentric wheel mounting devices for adjusting the axis of rotation of said guide wheels with respect to said track.

26. The apparatus of claim 25, and further comprising: an additional wheel supporting said carriage on said base plate on the side of said carriage opposite said track means.

27. The apparatus of claim 26, and further comprising: an additional adjustable eccentric wheel mounting device mounting said additional wheel on said carriage and adjustable to level said carriage.

28. The apparatus of claim 17, wherein said carriage comprises: a data record support including a glass plate.

29. The apparatus of claim 28, wherein said glass plate comprises: a mirror.

30. The apparatus of claim 28, and further comprising: a data record holding device on said data record support.

31. The apparatus of claim 30, wherein said data record holding device comprises: spring means.

32. The apparatus of claim 30, wherein said data record holding device comprises: a spring having a shape to engage at least one peripheral edge of the data record.

33. The apparatus of claim 30, wherein said data record holding device comprises: a spring having a shape to engage at least three peripheral edges of the data record.

34. The apparatus of claim 17, wherein: said light beam generating means and said beam directing means are both mounted on said rotatable member; and said carriage is mounted on said base plate.

35. The apparatus of claim 34, wherein said beam directing means comprises: beam deflecting means mounted and defining a beam path between said beam generating means and said beam adjustment means, said beam adjustment means including a lens.

36. The apparatus of claim 35, wherein said beam deflecting means comprises: mirror means.

37. The apparatus of claim 35, wherein said beam deflecting means comprises: prism means.

38. The apparatus of claim 35, wherein said beam deflecting means comprises: prism means for directing the light beam towards said lens; and mirror means for directing the light beam towards said prism means.

39. The apparatus of claim 34, wherein said beam generating means comprises: a source of coherent light.

40. The apparatus of claim 39, wherein said source comprises: a laser device.

41. The apparatus of claim 17, wherein said light beam receiving means comprises: a first opto/electronic transducer mounted on said rotatable member for converting the modulated light beam into first electrical signals; an electro/optical transducer mounted on said rotatable member and connected to said first

opto/electronic transducer for converting said first electrical signals into optical signals; an optical transmission line coupled to said electro/optical transducer and extending along the axis of rotation of said rotatable member; and a second opto/electronic transducer mounted coaxially of the axis of rotation of said rotatable member light coupled to said transmission line for converting said optical signals into second electrical signals.

42. The apparatus of claim 17, wherein said light beam receiving means comprises: a first opto-electronic transducer mounted on said rotatable member for receiving and converting the modulated light beam into first electrical signals; a pulse shaping circuit carried on said rotatable member and connected to said first opto/electronic transducer for reshaping said first electrical signals into second electrical signals; an electro/optical transducer mounted on the axis of rotation of said rotatable member and connected to said pulse shaping circuit for converting said second electrical signals into optical signals: and a second opto/electronic transducer mounted coaxially of the axis of rotation of said rotatable member light coupled to said electro/optical transducer for converting said optical signals into third electrical signals.

43. The apparatus of claim 17, wherein: said rotatably mounted member carries circuit components requiring first and second ranges of voltages; and shaft means mounting said rotatable member and including a slip ring structure for coupling said voltages of said first and second ranges between the fixed and rotating parts.

44. Apparatus for reading a data record on which the data is recorded on a data carrier in a

plurality of spaced arcuate data paths, comprising: a frame including a base plate and a top plate secured together spaced apart and substantially parallel to one another, and a wheel rotatably mounted between said base and top plates; a wheel drive motor coupled to said wheel for rotating the same; carriage means mounted on said base plate for linear movement radially with respect to the arcuate data paths, including data record support means for supporting a data record; skew correction means coupled to said data record support means and operable to correct skew of the data record with respect to the linear movement thereof; beam generating means mounted on said wheel for rotation therewith, said beam generating means including a laser device operable to emit a coherent light beam; beam directing means mounted on said wheel for rotation therewith, said beam directing means including lens means for directing a light beam in an arc towards and across the supported data record, beam deflecting means for receiving and deflecting the light beam from said beam generating means to said lens means, and radius correction means including means movably mounting said lens means and operable to move the same to correct the difference between the axes of rotation of said arcuate data paths and said wheel; beam receiving means for receiving the light beam after the same strikes and is modulated by the data record and converting the same into electrical signals representing the scanned data; and control means connected to said carriage means for incrementing said carriage means to said skew correction means for controlling skew correction, to said radius correction means for controlling radius correction, and to said beam receiving means for receiving and outputting said electrical signals.

45. The apparatus of claim 44, wherein said beam receiving means comprises: shaft means rotatably mounting said wheel, including passageway means therein extending to and along the axis of rotation thereof to and exiting at one end thereof; and signal transmission means for transmitting signals through said passageway means to said one end of said shaft.

46. The apparatus of claim 45, wherein said signal transmission means comprises: a fiber optic extending through said passageway means; and an opto-electronic transducer light coupled to said fiber optic and fixed to said top plate axially of said shaft means and connected to said control means.

47. The apparatus of claim 45, wherein said signal transmission means comprises: electrical leads extending through said passageway means; and electro/optical transducer connected to said electrical leads for converting electrical to optical signals; and an opto/electronic transducer light coupled to said electro/optical transducer and fixed to said top plate axially of said shaft means and connected to said control means.

48. Apparatus for reading digital data from a data record on which the data is recorded in a plurality of spaced data rows, comprising: a light source operable to emit a light beam; beam directing means mounted to receive the light beam and cyclically sweep the beam over a path including a section corresponding to the extent of a data row; a movably mounted carriage for supporting the data record in the area of the beam sweep path; control means connected to said carriage and operable to incrementally move said carriage between beam sweeps of said segment; and light receiving and conversion means mounted to receive the light beam after

the same strikes the data record and is modulated by the data and operable to convert the modulated light beam into electrical signals corresponding to the data.

49. The apparatus of claim 48, wherein: said light source comprises a laser device for emitting a coherent laser light beam.

50. The apparatus of claim 49, wherein: said laser device is fixed; and said beam directing means comprises a rotatable member and beam directing members mounted to direct the light beam along the axis of rotation of said movable member.

51. The apparatus of claim 49, wherein: said beam directing means comprises a rotatable member and beam deflecting members carried on said movable member; and said laser device is mounted on and rotates with said movable member.

52. A method of reading data from a data record which has data inscribed on a data carrier in spaced data rows, comprising the steps of: mounting the data record on a carriage; incrementally moving the carriage corresponding to the spacing of the data rows; generating a light beam; cyclically sweeping the light beam over the data record along the row path to modulate the light beam with the inscribed data; and receiving and converting the modulated light beam into electrical signals representing the data.

53. The method of claim 52, wherein the step of generating a light beam is further defined as: generating a coherent light beam.

54. The method of claim 52, wherein the step of cyclically sweeping the light beam is further defined as: rotating the light beam in a circle.

55. The method of claim 52, wherein the step of

cyclically sweeping is further defined as: directing the light beam over a path including a plurality of segments.

56. The method of claim 55, wherein the step of directing is further defined as: reflecting the light beam along parallel paths of different vertical position.

57. The method of claim 55, wherein the step of directing is further defined as: reflecting the light beam along a plurality of paths in horizontal and vertical planes.

58. The method of claim 52, wherein the step of receiving and converting is further defined as: shaping the electrical signals into defined pulses.

59. The method of claim 52, wherein the data record has a predetermined dimension in the direction of incremental movement, and further comprising the steps of: sensing changes in the predetermined dimension: and changing the length of the light beam path to compensate for dimensions differing from said predetermined dimension.

60. The method of claim 52, wherein the data record is to be incrementally moved with a predetermined orientation of the data rows, and further comprising the steps of: sensing the orientation of the data rows: and changing and correcting the orientation in response to sensing skew from said predetermined orientation.

61. A method of reading data from a data record which has data recorded on a data carrier in spaced arcuate data rows, comprising the steps of: incrementally moving the data record transversely of the data rows; projecting a light beam in a plane generally parallel to the plane in which the data record moves: rotating a deflection structure in the plane of the light beam to deflect the light beam to travel substantially

perpendicular to and be swept arcuately across the data record whereupon the data modulates the light beam; and receiving and converting the modulated light beam into electrical signals.

62. The method of claim 61, wherein the step of projecting a light beam is further defined as: generating a laser light beam and emitting the same in the plane generally parallel to the plane in which the data record moves.

63. The method of claim 61, and further comprising the step of: controlling the steps of incrementally moving the data record and rotating the deflection structure to provide incrementation equal to the data row spacing per revolution of the deflection structure.

64. A method of optically reading data from a data record which has data inscribed in a plurality of arcuate spaced rows having the same axis of rotation and borne on a substantially polygonal planar data carrier, comprising the steps of: rotating the data carrier about the axis of rotation of the arcuate data rows and in plane which is coplanar with its own plane; projecting a laser beam substantially perpendicular to the plane of rotation to strike and scan the data record to define a scanning arc corresponding to the arcs of the data rows; incrementally moving the scanning arc in steps of at least the spacing from one data row to the adjacent data row to modulate the laser beam with scanned data; and receiving and demodulating the modulated laser beam to recover the data.

65. An information record comprising: a body; and digital information carried on said body in arcuate rows, said rows of digital information having the same

radius and disposed in a spaced-apart nested arrangement.

66. The information record of claim 65, wherein: said body is a photographic film and said digital information comprises transparent portions of the film.

67. The information record of claim 65, wherein: said body is a photographic film and said digital information comprises opaque portions of the film.

68. The information record of claim 65, wherein: said body is a photographic film and said digital information comprises opaque-to-transparent and transparent-to-opaque transitions of the film.

69. The information record of claim 65, wherein: said body is substantially opaque; and said digital information comprises reflective areas on said body.

70. The information record of claim 65, wherein: said body is substantially reflective and comprising non-reflective areas on said body to provide reflective-to-non-reflective transitions representing the digital information.

71. The information record of claim 65, wherein: said body is substantially non-reflective and comprising reflective areas on said body to provide non-reflective-to-reflective transitions representing the digital information.

72. An information record comprising: a body of photographic film; and a plurality of arcuate rows of digital information borne by said body, said digital information comprising transitions between opaque and transparent areas and said rows being spaced and nested and having equal radii.

73. An information carrier comprising: a body of substantially non-reflective material, and reflective areas carried by said body in equal radii, arcuate,

nested rows and constituting at the transitions between reflective and non-reflective areas, digital information.

74. An information carrier comprising: a body of substantially reflective material, and non-reflective areas carried by said body in equal radii, arcuate, nested rows and constituting at the transitions between reflective and non-reflective areas, digital information.

75. An information record comprising a carrier of a material having a first light transmissive characteristic: rows of marks carried on said carrier, said rows being spaced apart and each of said rows extending in an arcuate manner and having the same radius as the other rows; and each of said marks having a second light transmissive characteristic which is different from said first light transmissive characteristic to therewith define digital data.

76. The information record of claim 75, wherein: said first light transmissive characteristic is greater than that of said second light transmissive characteristic, the differences defining the digital data.

77. The information record of claim 75, wherein: said second light transmissive characteristic is greater than that of said first light transmissive characteristic, the difference defining the digital data.

78. The information record of claim 75, wherein: the transitions between the first and second light transmissive characteristics of said carrier and marks constitutes the digital data.

79. The information record of claim 75, and further comprising: a header along one edge of the carrier, said header bearing visually perceptible information related to the digital data of said rows of marks.

FIG. 1

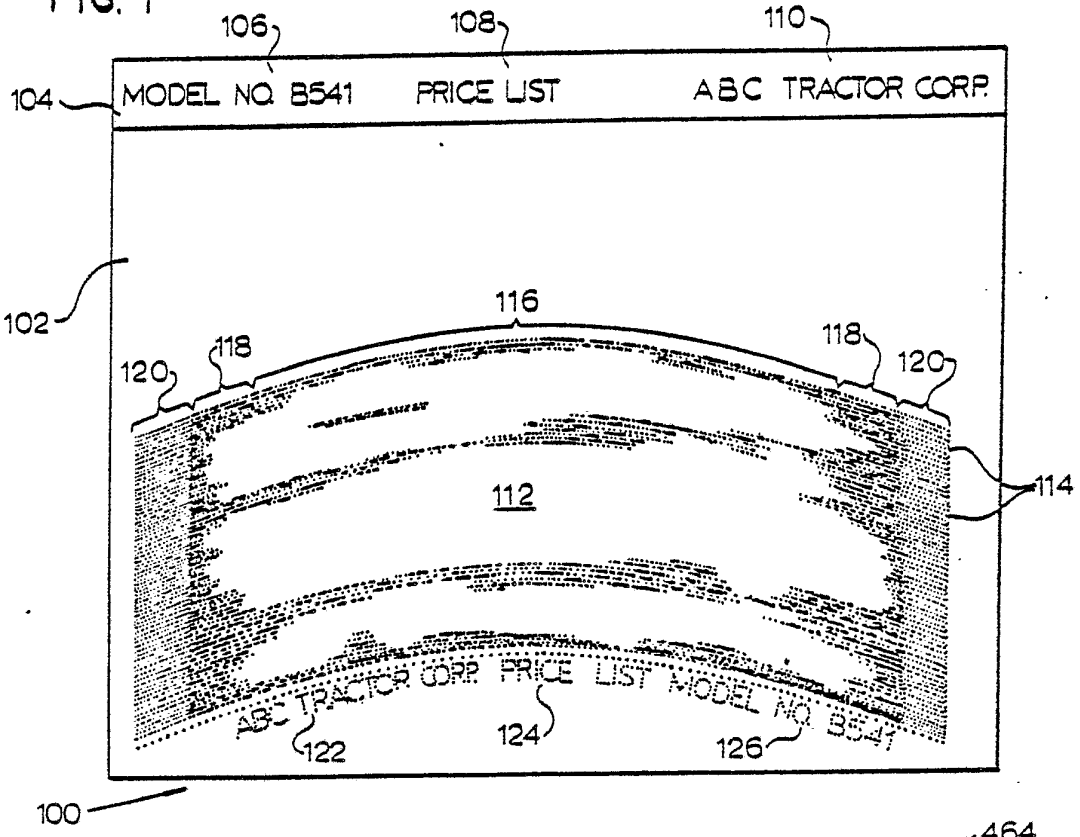
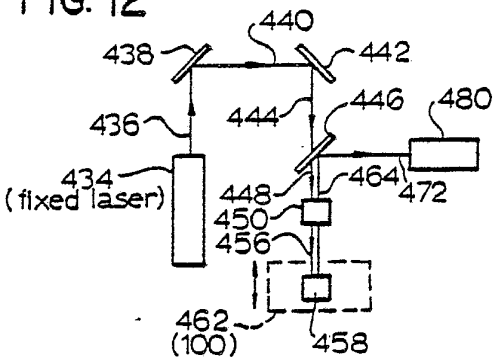


FIG. 12



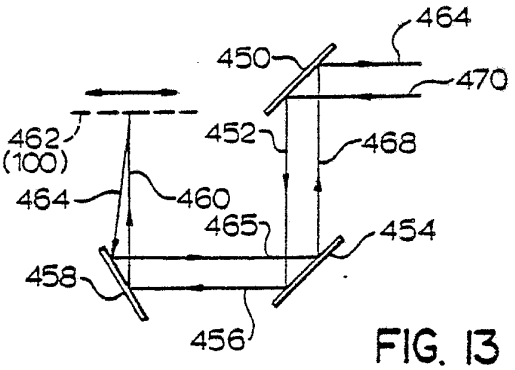
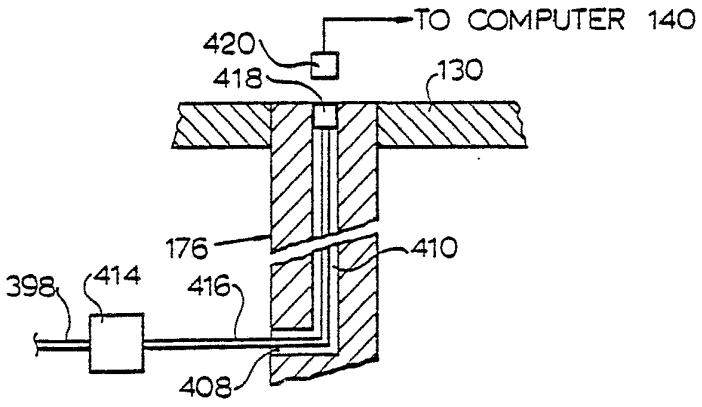


FIG. 13

FIG. 11



SUBSTITUTE SHEET

FIG. 2

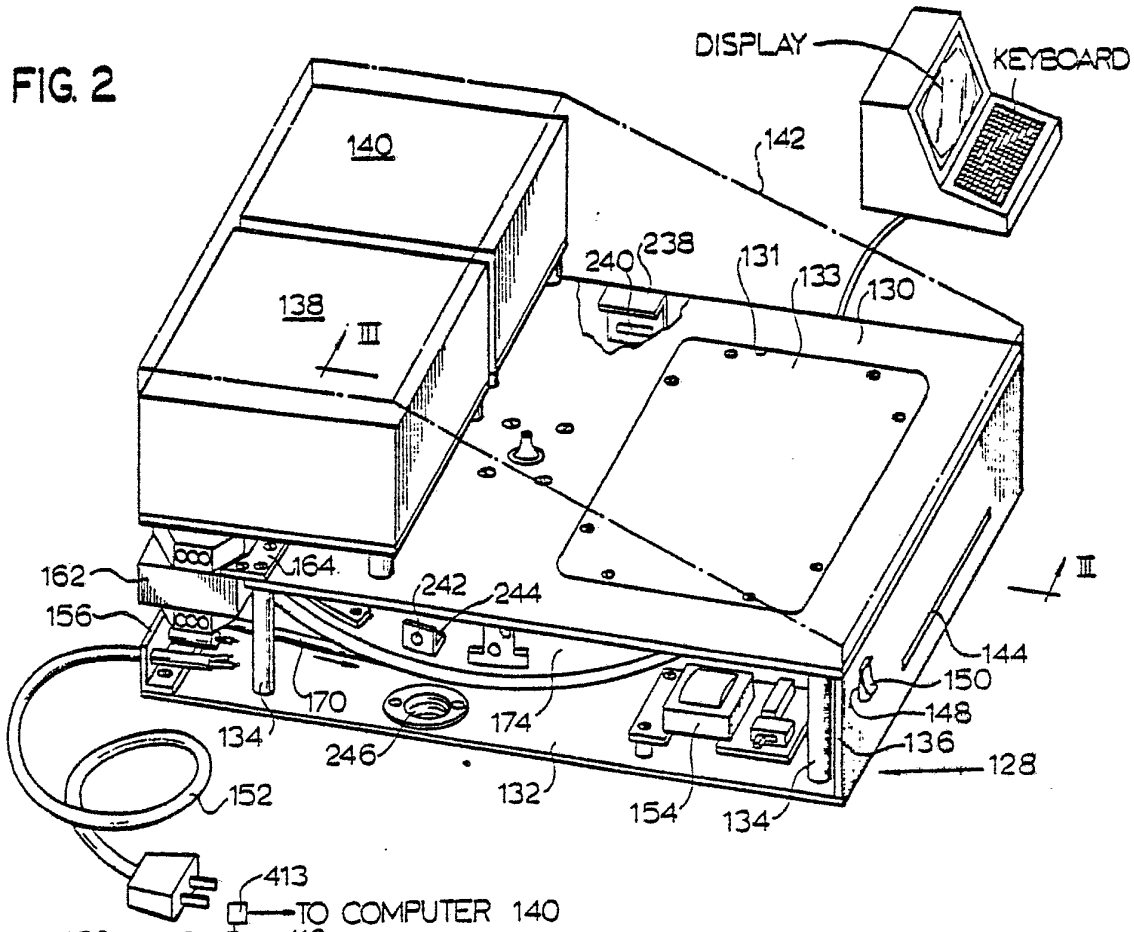


FIG. 3

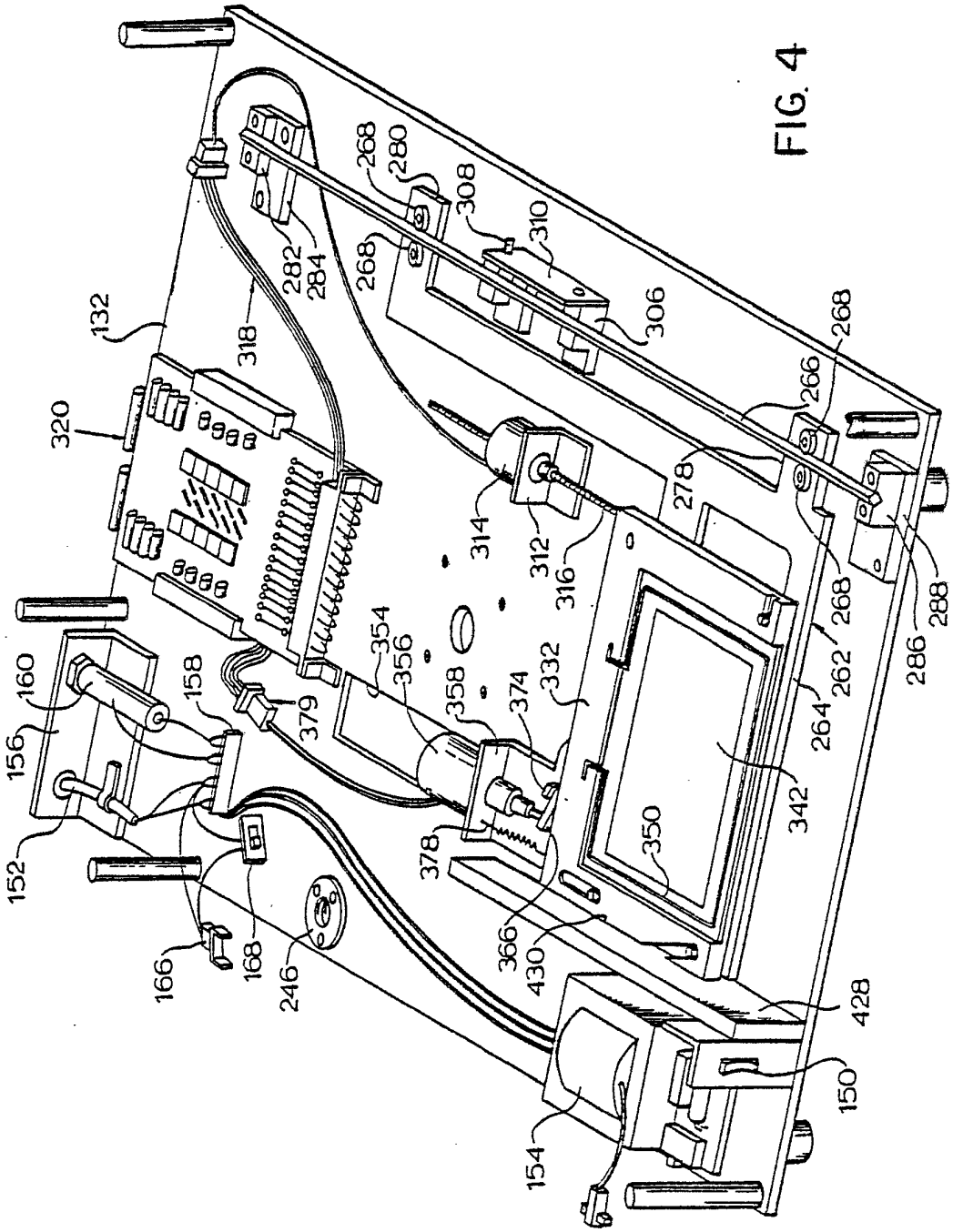


FIG. 4

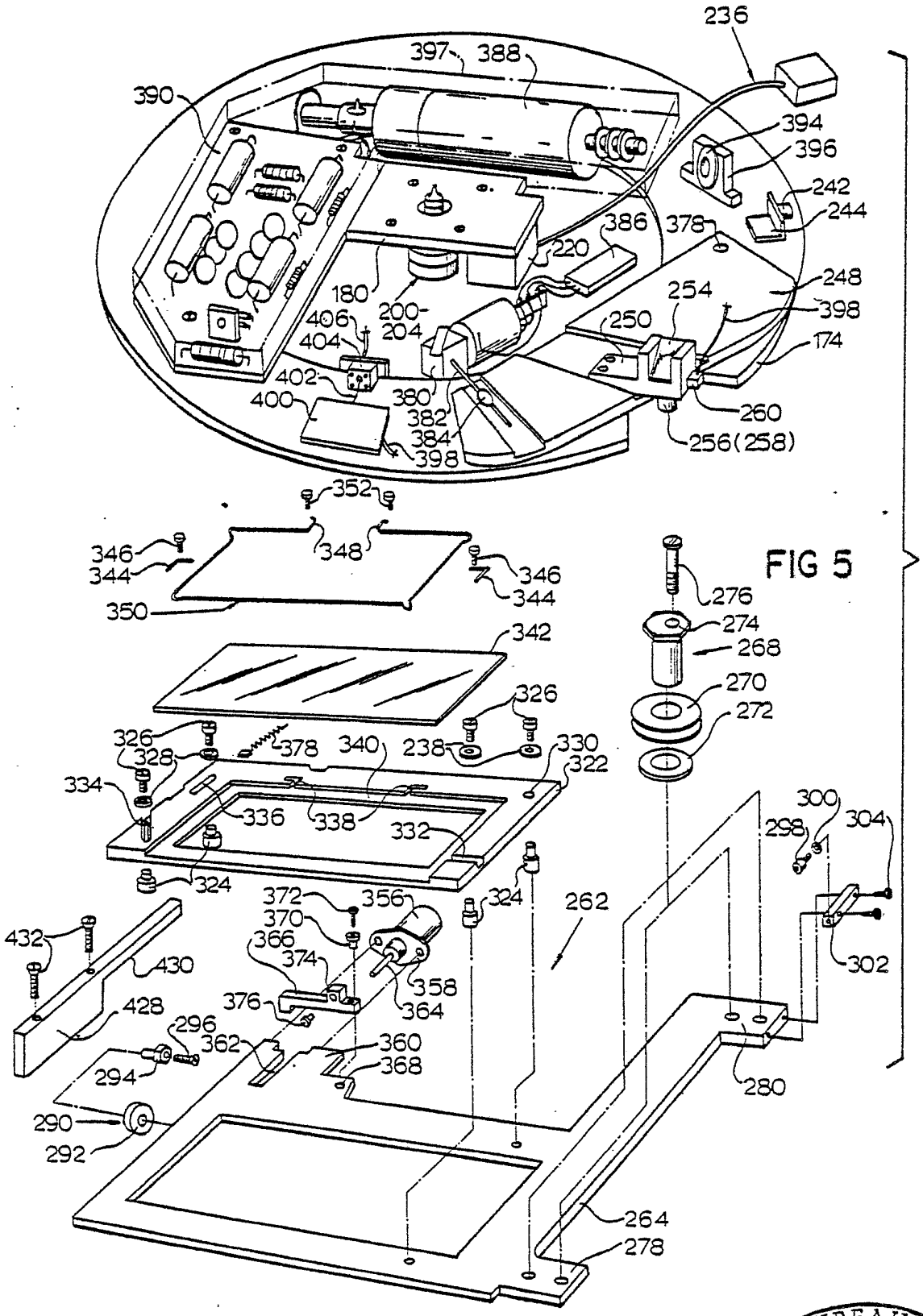


FIG 5

SUBSTITUTE SHEET

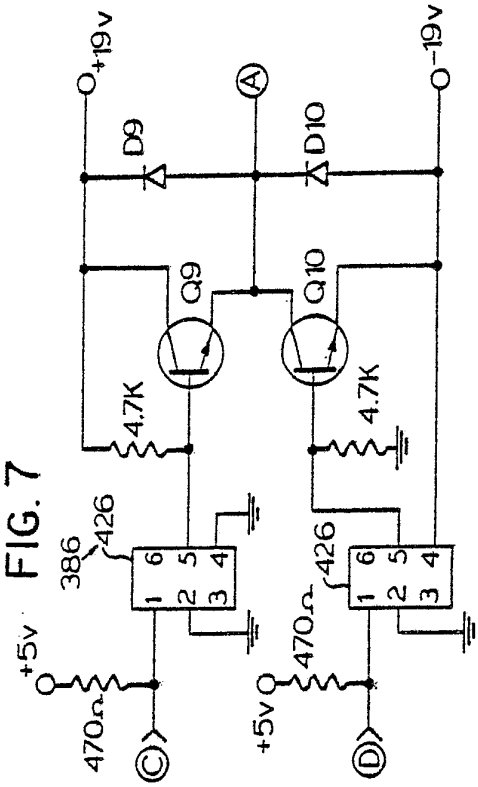


FIG. 7

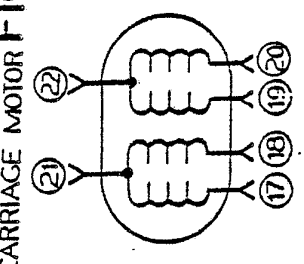


FIG. 8

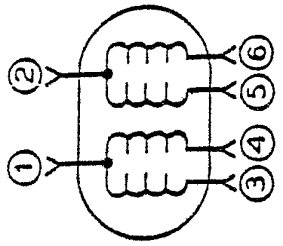


FIG. 9

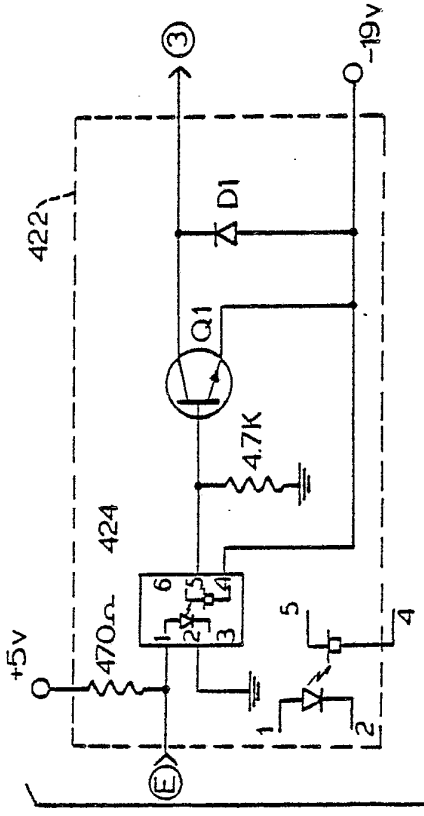


FIG. 6

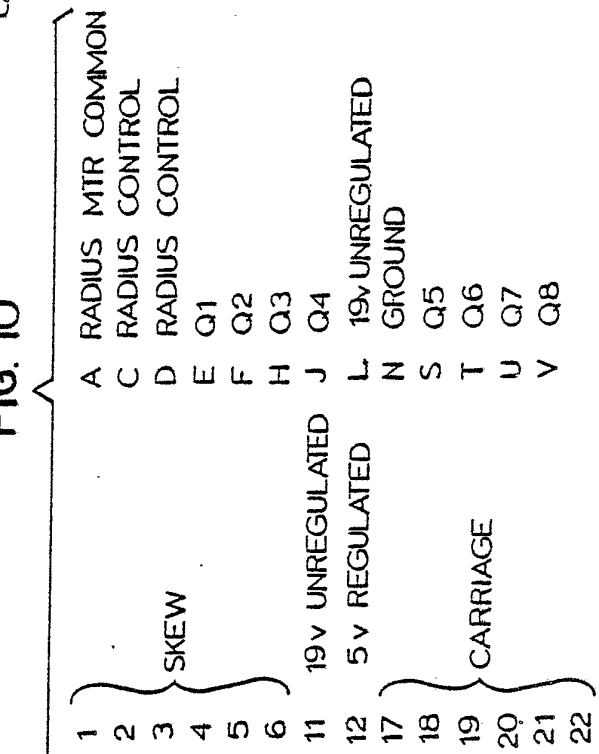


FIG. 10

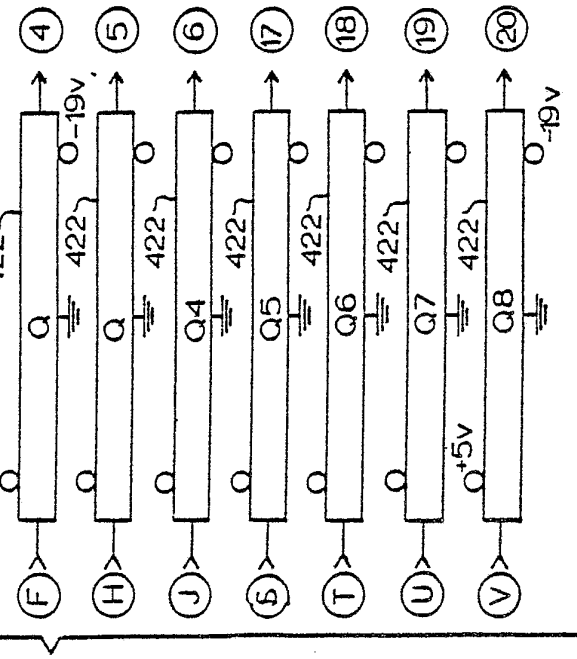


FIG. 6

SUBSTITUTE SHEET

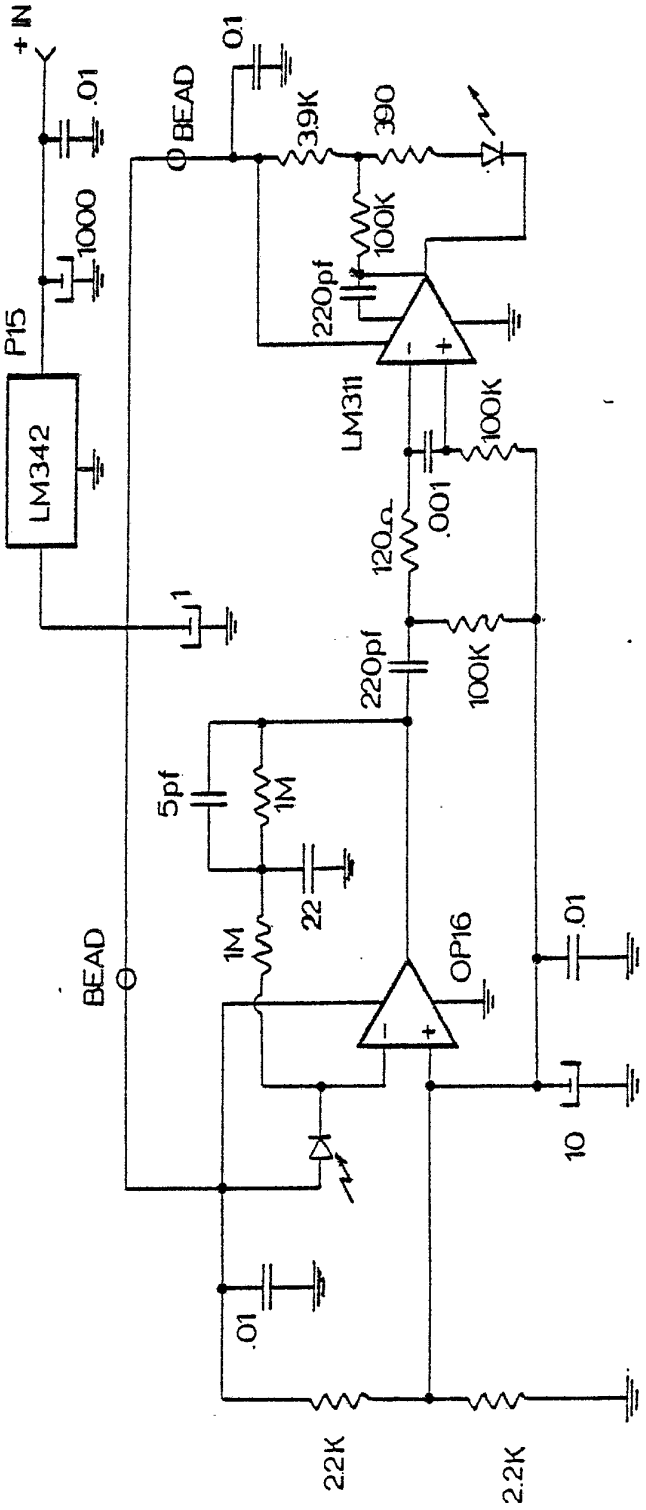


FIG. 14

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 81/01569

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) [3]

According to International Patent Classification (IPC) or to both National Classification and IPC

INT. CL.[3] G11B 7/00
U.S. CL. 369-44

II. FIELDS SEARCHED

Minimum Documentation Searched [4]

Classification System	Classification Symbols
U.S.	369-44,43,100,111,112,93,97,120,121,122,125,272,284,275, 369-288,273. 358-130,132,285,901,200,205,206, 365-120,127, 360-2,101,97,76,77, 250-202,236,235,234, 350-64,63,62,96,15

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [5]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [14]

Category *	Citation of Document, [16] with indication, where appropriate, of the relevant passages [17]	Relevant to Claim No. [18]
X	Laser Focus, Published August, 1979. See PP. 30,32,34.	1-79
X	US,A, 3,898,629, Published 5 August 1975, Westerberg.	1-79
X	US,A, 4,163,600, Published 7 August 1979, Russell.	1-79
X	US,A, 3,564,120, Published 16 February 1971, Taylor.	1-64
A	US,A, 4,136,261, Published 23 January 1979, Wada.	17-63
A	US,A, 3,765,743, Published 16 October 1973, Reaves.	16,37,38
A	US,A, 3,387,295, Published 4 June 1968, De Moss	8,60

* Special categories of cited documents: [15]

"A" document defining the general state of the art
"E" earlier document but published on or after the international filing date
"L" document cited for special reason other than those referred to in the other categories
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but on or after the priority date claimed
"T" later document published on or after the international filing date or priority date and not in conflict with the application, but cited to understand the principle or theory underlying the invention
"X" document of particular relevance

IV. CERTIFICATION

Date of the Actual Completion of the International Search [2]	Date of Mailing of this International Search Report [2]
15 January 1982	22 JAN 1982
International Searching Authority [1]	**Signature of Authorized Officer [20]**
ISA/US	Alan Faber

Form PCT/ISA/210 (second sheet) (October 1977)