A floppy disk drive including a pivotally mounted disk pack storing a plurality of floppy diskettes, a driver mechanism for rotating any one of the diskettes into a retrievable position, a picker mechanism for moving the one diskette between the retrievable position and an operative position at which the diskette is rotatable on a spindle, and a position control circuit for controlling the position of a recording head across the one diskette being in the operative position and for compensating for expansion, contraction, and eccentric rotation of the one diskette. The floppy diskettes are stored bent in the disk pack, but in the operative position the one diskette that had been retrieved from the retrievable position is stored only partially in the disk pack and parallel to the plane of rotation. A microprocessor controls the entire disk drive.

25 Claims, 19 Drawing Figures
FIGURE 16
(OPEN POSITION)

FIGURE 17
(CLOSED POSITION)
DISK DRIVE

TECHNICAL FIELD

This invention relates to information storage systems and, more particularly, to disk storage apparatus and drive control apparatus for disk drives.

BACKGROUND OF THE INVENTION

Information storage systems are employed to store data which are processed by data processors such as micro-computers, mini-computers and main frames. A disk drive is one type of information storage system whose use is rapidly expanding, in part because it provides high-capacity, on-line, random access storage for small to medium scale computer systems. The data storage medium of the disk drive can be a hard, i.e., inflexible, disk or a soft, i.e., flexible, disk or diskette, known as a floppy disk. The disk drive has a read/write recording head that is movable radially across concentric data tracks on the diskette while the diskette is rotated on a spindle. Disk drives are identified by their sizes and given generic names such as mini-drives and maxi-drives.

As can be appreciated, a wide variety of technical problems must be satisfactorily solved before a disk drive can be adequately utilized. One problem is to provide a disk drive of a given size with a higher, online, data storage capacity. One type of disk drive solves this problem by providing a disk pack which stores a plurality of diskettes. A drive mechanism linearly moves the disk pack up and down to bring any one of the diskettes into a retrievable position. The one diskette then is retrieved or picked from the disk pack and moved inwardly to an operative or rotational position on the spindle. After use, the diskette is returned to the disk pack and the procedure can then be repeated to retrieve another diskette.

While the storage of a plurality of diskettes in the disk pack increases the data storage capacity of the disk drive, the linear up and down motion of the pack adds to the space requirements of the disk drive. Furthermore, a relatively complicated drive mechanism is utilized to move the disk pack up and down. Also, this space requirement and complicated drive mechanism increase the cost of the disk drive.

Another problem relates to the design of a picker mechanism for retrieving the diskette from, and returning it to, the disk pack. A prior picker mechanism has spring-biased jaws which are driven onto an edge of the diskette to open slightly and clamp the diskette. Then, the jaws hold the diskette by friction while the diskette is moved to the operative position. Thereafter, the jaws slide off the diskette to unlamp or release the diskette in the operative position. One disadvantage is that the jaws are clamped on the jacket of the diskette, which can cause wear of the jacket and damage to the data storage medium in the jacket. Also, relatively high electrical power and a large motor are required to drive the jaws onto the diskette edge. Furthermore, the power needed to drive the jaws onto the edge must be balanced against the power used in moving the jaws towards the operative position to prevent the jaws from slipping off the edge.

Yet another problem is the expansion or contraction exhibited by the data storage medium in response to temperature and humidity conditions. This expansion or contraction increases or decreases the distance between centerlines of the concentric data tracks, causing problems in accurately positioning or servioing the recording head over a data track. One prior drive uses an outer and an inner servo track being outside and inside the data tracks, respectively. The expansion and contraction of the data storage medium are compensated by measuring the variable distance between the inner and outer servo tracks, and then assigning proportional locations for all the data tracks which lie between the servo tracks. As the recording head is moved to a particular data track, a scale is sensed to detect the distance moved by the head. When the distance moved by the head indicates that the assigned position has been reached, the head is assumed to be centered over the particular data track and movement of the head is stopped.

One disadvantage with the above compensation scheme is that two servo tracks must be written on the storage medium by the diskette manufacturer. The use of two servo tracks also reduces the available data storage capacity of the diskette. Another disadvantage is that the recording head must be moved across the diskette between the inner and outer servo tracks to detect their relative positions and calculate the distance between them, thereby requiring a relatively long time for collecting the data to make the compensation. Still another disadvantage is the use of a complicated algorithm for computing the data assigning positions to the data tracks.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a novel information storage system.

It is another object of the present invention to provide a disk drive of given size and space requirement with high storage capacity.

Yet another object of the present invention is to provide a picker mechanism having mechanically actuated jaws.

Still another object of the present invention is to provide an improved technique for compensating for expansion or contraction of a data storage medium.

The above and a number of other objects of the present invention are obtained through the use of an information storage system, such as a disk drive, that has a pivotal means for storing a plurality of data storage devices, such as diskettes, means, including a mechanically actuated picker mechanism, for picking and moving one of the data storage devices from a retrievable position of the pivotal storing means to an operative position at which the one data storage device is used, and a position control circuit for positioning a recording head relative to the one data storage device at the operative position. The position control circuit compensates, for example, for expansion or contraction of the data storage medium by having a scale which expands and contracts proportionally with the data storage medium to provide information for locating the expanded or contracted data tracks on the data storage device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a simplified, partially broken view, in perspective, of a disk drive.

FIG. 2 is a top view of the disk drive of FIG. 1.

FIG. 3 is an exploded view, in perspective, of a pivotal storage box and a drive for pivoting the box.
and an open front end 40 which faces the spindle 22. Support tray 32 which is pivotally mounted on base 14 an open rear end 36 that can be closed by a pivotally mounted door 38, connected to base 14 (see FIG. 1), and an open front end 40 which faces the spindle 22.

FIG. 4 is a section taken along lines 4-4 of a disk pack of the pivotal storage box of FIG. 3. FIG. 4A is a section taken along lines 4A-4A of FIG. 4.

FIG. 4B is a top view, partially broken away, of the disk pack shown in FIG. 3.

FIG. 5, FIG. 6 and FIG. 7 are illustrations used to explain the operation of the pivotal storage box.

FIG. 8 illustrates, in perspective, a picker mechanism for the disk drive of FIG. 1.

FIG. 9, FIG. 10 and FIG. 11 are illustrations of the picker mechanism, partially broken away, used to explain its operation.

FIG. 12 is a top view of a data storage medium of a diskette used with the disk drive of FIG. 1.

FIG. 13 shows a signal waveform used to explain a recording head position control circuit utilized by the disk drive.

FIG. 14 is a block diagram of the position control circuit.

FIG. 15 is a perspective view of a disk clamp mechanism in combination with the picker mechanism.

FIG. 16 and FIG. 17 are views used to explain the operation of clamping and reclamping a diskette on the spindle in the operative position.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 and FIG. 2 illustrate an information storage system 10. The system 10 is, for example, a disk drive 12 and, in particular, a floppy disk drive. While the present invention will be described in connection with the floppy disk drive 12, it will become apparent that a number of the components to be described can be utilized for other information storage systems in general and other types of disk drives in particular.

Disk drive 12 has an envelope or base 14 which mounts pivotally a box 16 that stores or houses a plurality of data storage devices 18 such as floppy disks or diskettes. A picker mechanism 20 is supported on base 14 to pick or clamp, and move, any one of the floppy diskettes 18 stored in box 16 between a retrievable position into which the one diskette 18 is pivotable and an operative position at which the one diskette 18 is rotatable on a spindle 22.

A position control device 24, connected to the base 14, includes a carriage 26 which carries and moves a recording head 28 radially inwardly and outwardly across the diskette 18 being rotated on the spindle 22. Position control device 24, as will be further described, accurately positions the head 28 with respect to concentric data tracks on a data storage medium (described below) of the diskette 18 after compensating for expansion or contraction of the data storage medium due to, for example, temperature and humidity conditions, or after compensating for eccentric rotation of the data storage medium due to slight misalignment on the spindle 22.

Base 14 also has a straight edge guide 30 which supports and guides a diskette 18 during its movement between the retrievable position and the operative position. As shown in FIG. 3, the pivotable box 16 includes a support tray 32 which is pivotally mounted on base 14 by a pair of pivots at 34 (see also FIG. 1). Tray 32 has an open rear end 36 that can be closed by a pivotally mounted door 38, connected to base 14 (see FIG. 1), and an open front end 40 which faces the spindle 22. Tray 32 also has a flag 41A that is movable into a fixed optical sensor 41B to signal that box 16 is in a home position, as will be further described.

Box 16 also includes a disk pack 42 which is insertable into the tray 32 through the rear end 36 and rotates with tray 32 about pivots 34. Disk pack 42 also has an open rear end 44 which can be closed by door 38 and an open front end 46 facing spindle 22. Disk pack 42 has a guide 48 and tray 32 has a pin 50 which locates or centers the disk pack 42 via guide 48 upon insertion of the pack 42 into the tray 32. Disk pack 42 also has a spring-biased detent 52 that is biased into a pack groove 54, as well as a cut-out 56, for purposes to be described.

Disk pack 42 removable stores or houses a plurality of the diskettes 18. As illustrated in FIG. 4, FIG. 4A, and FIG. 4B, a side 58 of disk pack 42 has a plurality of guides or rails shown generally at 60 for supporting the plurality of diskettes 18, respectively. As one example, there are five guides 60-1, 60-2, 60-3, 60-4, 60-5, in which four are angled, e.g., V-shaped, and one, i.e., guide 60-3, is straight. The diskettes 18, which are floppy diskettes and, therefore, flexible, are bent or follow the angle of guides 60 when supported on these four guides. The diskette 18 supported on guide 60-3 is straight since this guide is straight.

Each of the guides 60-1, 60-2, 60-4 and 60-5 has a forward, substantially straight guide portion 62-1, 62-2, 62-4 and 62-5 joining a rearward, substantially straight guide portion 64-1, 64-2, 64-4 and 64-5 via an apex 66-1, 66-2, 66-4 and 66-5. The guide portions 62-1, 62-2, 62-4 and 62-5 are angled such that their centerlines 68-1, 68-2, 68-4, 68-5 extend through a pivot point 69 on side 58 coincident with the pivots 34. Guide 60-3, which is not angled, also has a centerline 68-3 extending through point 69. Furthermore, the apices 66-1, 66-2, 66-4 and 66-5 are approximately one-third in from the front end 46.

This angling of the guide portions 62-1, 62-2, 62-4 and 62-5 and location of the apices 66-1, 66-2, 66-4 and 66-5 provide a number of advantages. The diskettes 18, when supported on the plurality of guides 60, are “fanned” at the rear end 44 and front end 46. This provides sufficient space between the diskettes 18 to remove a particular diskette by hand from the rear end 44 of disk pack 42 and, more importantly, to retrieve a particular diskette from the front end 46 of disk pack 42 by picker mechanism 20. Also, there is sufficient spacing between the plurality of guides 60 in the area of the apices 66-1, etc., to provide, for example, as many as five such guides 60 within the given height of disk pack 42. Moreover, for reasons which will become apparent, the distance between a given apex 66-1, etc., and the axis of rotation of spindle 22 is equal to the radius of the diskettes 18. The diskettes 18 can be, for example, mini-floppy diskettes having a diameter of 514. Still furthermore, and also as will become apparent, the disk pack 42 need only be rotated a limited amount around pivots 34 and point 69 to move any one of the diskettes 18 into the retrievable position for picking by picker mechanism 20.

With reference again to FIG. 3, base 14 supports a drive mechanism 70 for rotating box 16 about pivots 34. Mechanism 70 includes an index cam 72, having a cam surface 74, which is connected to the bottom of tray 32, and a clevis 76 having a cam follower 78 which moves along surface 74. A stepper motor 80 and lead screw 82 are connected to clevis 76 to drive cam follower 78.
along cam surface 74, thereby pivoting tray 32 and, hence, disk pack 42.

Stepper motor 80, as will be further described, is selectively controlled to drive or move cam follower 78 to any one of five discrete positions along cam surface 74. This action brings any one of the diskettes 18 into the retrievable position. FIG. 5 illustrates the bent guide 60-1 of box 16 as being rotated downwardly to a position called the retrievable position. In this retrievable position, the diskette 18 is guided by guide 60-1 and by edge guide 30 on base 14. In the operative position, a portion of the guide portion 62-1 is still supported in guide portion 62-1. Moreover, the guide portion 62-1 is parallel to the plane of rotation of diskette 18. This, thus, while the diskette 18 is flat or angled when in the retrievable position, it is flat when in the operative position.

It may now be appreciated that box 16 can be pivoted to any one of the five diskettes 18 into the retrievable position shown in FIG. 5. The angles of guide portions 62-1 through 62-5 are such that when any one of them is in the retrievable position, the corresponding diskette 18 is moved into the operative position, the diskette will be flat having the parallel plane of rotation.

After using the diskette 18 in the operative position, the picker mechanism 20 returns this diskette 18 to the retrievable position. Then, box 16 can be pivoted to any one of the other rotatable positions to move another diskette 18 into the retrievable position, which is then brought into the operative position. FIG. 7 shows another diskette 18 being in such an operative position. This pivoting of box 16 is random in the sense that box 16 can be pivoted from any one of the five discrete positions to move any one of the five diskettes 18 into the retrievable position shown in FIG. 5. The angles of guide portions 62-1 through 62-5 are such that when any one of them is in the retrievable position, the corresponding diskette 18 is moved into the operative position, the diskette 18 is flat when in the retrievable position, and the corresponding diskette 18 is flat when in the operative position.

A drive mechanism 140 for moving the carriage 96 along guide rods 98 includes a stepper drive motor 142 which drives a continuous belt 144 that is coupled to a drive mechanism 140 for moving the carriage 96 along guide rods 98. As will be further described, this selection process is performed under program control of a data processor, such as a microprocessor, which controls energization of stepper motor 80.

FIG. 8 illustrates the picker mechanism 20 in relation to a diskette 18 that is assumed to be in the retrievable position. The diskette 18 has a jacket 84 surrounding a disk-shaped data storage medium 86. The jacket 84 of each diskette 18 has a slot 88 that is aligned with cut-out 56 of disk pack 42 and has an edge groove 90 (see FIG. 8) that is aligned with spring biased detent 52 and groove 54 of disk pack 42 when the diskette 18 is stored in the pack 42.

Picker mechanism 20 includes a pick head 92 that has a pair of openable and closeable jaws 94 which pick or clamp the diskette 18 via slot 88. Pick head 92 is supported on a carriage 96 which rides on a pair of guide rods 98. A spring 100 is disposed between the upper of the jaws 94 and the carriage 96 to bias the jaws 94 closed. The upper of the jaws 94 moves vertically on a shaft 102 that has a cam follower 104 connected across and extending away from shaft 102.

FIG. 5 illustrates the bent guide 60-1 of box 16 as being rotated downwardly to a position called the retrievable position. FIG. 6, the picker mechanism 20 has retrieved the diskette 18 from guide 60-1 and moved it to the operative position at which the diskette 18 is clamped on spindle 22 and ready to be rotated. In moving from the retrievable position to the operative position, the diskette 18 is guided by guide 60-1 and by edge guide 30 on base 14. In the operative position, a portion of the diskette 18 is still supported in guide portion 62-1. Moreover, the guide portion 62-1 is parallel to the plane of rotation of diskette 18. This, thus, while the diskette 18 is flat or angled when in the retrievable position, it is flat when in the operative position.

A support member 106 supports a pick cam 108 near the retrievable position. Pick cam 108 has a lever 110 that is pivotally connected to support member 106 by a pivot 112 and is biased into the position shown by a spring 114. A cam surface 116, which is slanted and fixedly connected to lever 110 and in the path of cam follower 104, has an incline 118, an incline 120, and an underside 122.

Support member 106 also supports a release cam 124 near the operative position. Release cam 124 has a lever 126 that is pivotally connected to support member 106 by a pivot 128 and is biased into the position shown by a spring 130. A cam surface 132, which is slanted and fixedly connected to lever 126 and in the path of cam follower 104, has an incline 134, an incline 136, and an underside 138.
126 upwards and bypass release cam 124 without opening jaws 94. As shown in FIG. 11, as this disk-out movement continues, closed jaws 94 push on edge 150 via depending member 147 to move diskette 18 towards the retrievable position. As cam follower 104 then rides along incline 118 to begin opening jaws 94, which now lose contact with edge 150, carriage 96, and hence pick head 92, the outer, further, until diskette 18 is moved to retrieve diskette 18. The manner in which cam follower 104 is moved beyond cam surface 116 to retrieve diskette 18 and then moved only onto incline 118 to return diskette 18 to the retrievable position will be described below.

Then, at the position of cam follower 104 at which it is stopped on incline 118, groove 90 in diskette 18 starts to receive spring-biased detent 52 on disk pack 42. Detent 52 then rides into groove 90, forcing or drawing diskette 18 into the retrievable position. In addition to this function, detent 52 also maintains and aligns all of the diskettes 18 in the disk pack 42. As with stepper motor 80 which is used to pivot box 16, stepper motor 142 is energized under program control by the data processor to be described below.

FIG. 12 illustrates the data storage medium 86 of the diskette 18 that is used in conjunction with the position control device 24. The storage medium 86 is made of a material, such as Mylar, which will expand or contract in response to various temperature and humidity conditions. Another factor to be more fully discussed is that the diskette 18 may be slightly mis-clamped on the spindle 22 in the operative position such that the storage medium 86 may rotate slightly eccentrically.

The diskette 18 is manufactured or used with a reference datum shown generally at 152. As one example, the reference datum 152 is a single concentric reference or servo track 154. The reference track 154 has single frequency signal of, for example, 125 KHz written on it and is 0.0012" or 12 mils wide. Also included on the storage medium 86 for position control purposes are a plurality of sector holes 156, e.g., sixteen equiangularly spaced sector holes, although only every other or eight of the holes 156 need be utilized for position control purposes. An index hole 158 is also made on the storage medium 86 for indexing purposes. Storage medium 86 also has, for example, concentric data tracks T0-T153 on which to record data, each track T0-T153 being, for example, 0.0024"±0.0002" wide with a center-to-center spacing of 0.0059" at 70° F. and 50% relative humidity. Among other things, expansion and contraction of the medium 86 will increase and decrease the center-to-center distance of the data tracks T0-T153, as well as move outwardly and inwardly the inner diameter (I.D.) of reference track 154.

FIG. 13 shows a curve or waveform 160 of the amplitude vs. width of the single, as outputted by the recording head 28, frequency signal from the inner diameter (I.D.) of the data storage medium 86. Storage medium 86 is shown on the spindles 22 which can be rotated by a motor drive shown generally at 168. Also shown is the carriage 26, which is driven by a motor 170, such as a 1.5", 4-phase, stepper motor, and the recording head 28.

Carriage 26 also carries a sensor 172 which is of a material that expands and contracts proportionally to the material of data storage medium 86 in response to temperature and humidity conditions. Preferably, sensor 172 is of the same material as the medium 86 so that if the latter is Mylar, then the former is Mylar. Sensor 172 has a readable scale 174 which constitutes a plurality of spaced apart scale lines or indicia 176. The spacing of scale lines 176 is such that there are 10 lines or pulses (pips) corresponding to the center-to-center distance between each adjacent data track T0-T153, and 20 lines or pulses between reference position RP of reference track 154 and track T0. Sensor 172 also has a carriage home marker 178, which indicates when the carriage 26 is in a home position. In this home position, as shown in FIG. 12 and FIG. 14, recording head 28 should always be substantially outside the O.D. of reference track 154. This home position is predetermined to account for worst case conditions in which the medium 86 may expand, causing the reference track 154 to move outwardly, or in which slight eccentric rotation of the medium 86 on spindle 22 may cause portions of reference track 154 to move outwardly towards head 28 in the home position.

A microprocessor 180, such as the Intel 8051, manufactured by Intel Corporation, Santa Clara, California, receives a home position signal from a fixed sensor 182 via a line 184, an amplifier 186 and a line 188. Sensor 182 is at the fixed position to sense marker 178 when carriage 26 is moved into the home position. A sector/index sensor 190, having a light source 192 and light sensor 194 disposed on opposite sides of medium 86, generates sector signals and index signals as sector holes 156 and index hole 158 rotate between source 192 and sensor 194. These sector and index signals are fed over a line 196, an amplifier 198 and a line 200 to an interrupt port of microprocessor 180. Recording head 28 produces an amplitude signal, as shown in FIG. 13, proportional to the amplitude of the frequency signal on reference track 154. This signal is coupled to an R.F. amplifier 202 over a line 204, whose output on a line 206 is coupled to a rectifier and peak detector 208 that generates a DC voltage signal proportional to the amplitude of the frequency signal. An A/D converter 210, under control by microprocessor 180 over a control line 212, converts the DC voltage output of detector 208 on a line 214 to a digital number which is fed on a data bus 216 to microprocessor 180.

A fixed quadrature scale sensor 218 senses each scale line 176 on scale 174 as the latter moves radially across
medium 86. Sensor 218 produces four pulse signals on four lines 220 each time a scale line 176 moves outwardly or inwardly across sensor 218. Logic in a directional signal and pulse detector 222, in response to the information on lines 220, produces an outward/inward directional control signal on a line 224 and a pulse signal on a line 226 for each scale line 176. An up/down counter 228 is switched into an up-count state in response to the outward directional control signal on line 224 or a down-count state in response to the inward directional control signal on line 224, and counts each pulse on line 226. Under control by microprocessor 180 via a control line 230, counter 228, which may be a 16-bit counter, sends the count over data bus 216 to microprocessor 180.

As previously indicated, motor 170 is a 1.8°, 4-phase stepper motor which would move carriage 26 and hence head 28 a distance of 0.0059 inches for each full step of 1.8°. A motor control circuit shown generally at 232 includes a driver circuit 234 that controls motor 170 over four lines 235 by sourcing and sinking current to the two sets of coils (not shown) in motor 170. Driver circuit 234 receives a conventional 4-phase drive signal from microprocessor 180 over two lines 236. Given the four possible states of lines 236, four possible states of current flow directions exist for the two coils in motor 170. Motor 170 is locked at even stepped increments when a steady state condition exists with lines 236 fixed in any one of their four possible states. Motor 170 moves in full step increments given a change in state in one of the two lines 236 which results in corresponding changes in state of current flow to the motor coils over lines 235.

Under the control of a timer 240 and gating signals it generates on two lines 238, full step movements in motor 170 are divided by 100, such that corresponding movements in head 28 are 0.5 microinches of 1/100th of center-to-center tracks spacing. In particular, microprocessor 180 generates data words on data bus 216 and loads these into timer 240 with control line 242. These data words provide information for producing gating signals on lines 238 which result in driver circuit 234 driving current on lines 235 in a phase modulated fashion in one of the coils of motor 170 in a duty cycle ranging from 0 to 100%.

In the overall operation of position control circuit 166, under software control to be further described below, microprocessor 180 first activates motor control circuit 252 to drive motor 170 to move carriage 26 and hence head 28 in the outward direction towards the home position. When this home position is reached, sensor 182 detects marker 178, thereby to signal microprocessor 180 via amplifier 186. Microprocessor 180 then controls circuit 232 to deenergize motor 170 and stop the movement of carriage 26. During this outward movement, scale 174 has been moving across quadrature scale sensor 218. Thus, counter 228 has been placed in an up-count state and has been incremented with each passage of a scale line 176 across sensor 218 to store a count corresponding to this home position.

Thereafter, microprocessor 180 activates control circuit 232 to energize motor 170 to drive carriage 26 and head 28 in the inward direction towards reference track 154. During this movement, counter 228 is in the count-down state and has been decremented with each passage of a scale line 176 across sensor 218.

Assume now that head 28 has crossed the O.D. of reference track 154 and is moving across reference track 154 to produce level amplitude portion 162 of waveform 160. At each step of motor 170, and for one revolution of data storage medium 86, microprocessor 180 receives 8 sector signals via sensor 190 at its interrupt port to receive and store 8 data words from A/D converter 210 identifying the amplitude F at 8 corresponding circumferential locations of reference track 154. This sequence of stepping motor 170 and hence head 28 across level portion 162 and collecting 8 data words from A/D converter 210 at each step continues as head 28 approaches the I.D. of reference track 154 to produce sloping portion 164. If it is assumed that the amplitude F is constant across level portion 162 at each of the 8 circumferential locations, microprocessor 180 now identifies this as the reference track 154 and stores the value F for each of these 8 circumferential locations for later use.

Then, under control of microprocessor 180 and control circuit 232, head 28 is microstepped to produce sloping portion 164 and at each step the above-described sequence of obtaining 8 amplitude values or data words of the amplitude of waveform 160 is performed. Assume now, for example, that head 28 is at a position along portion 164 such that all 8 amplitude values are within the range <0.8F and > a value 5, i.e., the amplitude of the frequency signal along 8 circumferential positions is within this range. The value 5 represents a noise factor so that if an amplitude is detected that is equal to or less than this value, a reclamping occurs of diskette 8 on spindle 22, as will be described in connection with FIGS. 15-17. Also assume that data storage medium 86 is not rotating eccentrically and that if there is expansion or contraction it is uniform about the circumference. Finally, also assume that the value 0.5F corresponds to an absolute reference position RP for subsequently accurately locating the data tracks T0-T13.

Accordingly, at the present position of head 28 along portion 164, there are amplitude ratiot0-t17 at the 8 corresponding circumferential points on track 154 that are identical, these rati being calculated by microprocessor 180 and equal to the amplitude along portion 164 divided by the previously determined amplitude F of level portion 162 at the 8 circumferential positions. Microprocessor 180 can then calculate the offset of head 28 at the 8 circumferential positions corresponding to t0-t17 from the absolute reference position corresponding to 0.5F. Moreover, a single count N presently in counter 228 for these 8 sectors identifies the present position of head 28.

More particularly, assume that the abovementioned offset of the position of head 28 from 0.5F corresponds to a distance equal to a count of 1 counted by counter 228. Microprocessor 180 then can calculate the position of data track T0 (the outer data track), since it is predetermined that this track T0 is 20 pips or scale lines 176 from the reference position RP of 0.5F. That is, microprocessor 180, having the count N, can then subtract 1 due to the offset and then subtract 20 to store the number representing the position of track T0 relative to this reference position RP. Thereafter, microprocessor 180, in the manner previously described, can drive head 28 in the inward direction with scale 174 crossing sensor 218. When counter 228 has counted down 21 pips, 1 pip for the offset and 20 pips for the spacing between the reference position RP and the separation distance to track T0, the count in counter 228 equals the number calculated by microprocessor 180. Microprocessor 180
can now stop the head 28, which will be centered on data track T₀. This, it can now be appreciated, is because sensor 174 and scale 176 have expanded or contracted the same amount as data storage medium 86. That is, there always are 20 pips between the reference position RP of reference track 154 and data track T₀, though the spacing between pips will increase or decrease, as will the spacing between these two tracks, with expansion or contraction. Note also that there will always be 10 pips between the centerlines of adjacent data tracks T₀–T₁₅₃.

A number of other factors can now be appreciated. The offset for each of the 8 sectors when count N occurs is individually determined. Thus, if the rotation of medium 86 is slightly eccentric, different offset values are calculated and stored for the 8 sectors. As the medium rotates, microprocessor 180 will move head 28 in accordance with the count in counter 228 to cause head 28 to follow the eccentricity of track T₀. Also, head 28 can be aligned with any other data track T₁–T₁₅₃ by moving carriage 26 until counter 228 has counted down by 10 for each data track to be crossed in seeking a new data track. A similar procedure occurs when moving head 28 from an inner data track to an outer data track, except counter 228 now counts up by 10 for each data track that is crossed.

The data storage medium 86 has been described as including data tracks T₀–T₁₅₃ as well as reference track 154. The position control circuit 166 has been described as functioning to move and align recording head 28 over any one of the data tracks T₀–T₁₅₃ relative to the reference track 154. However, it will be appreciated that the principles of this alignment technique can be employed with other disk drives in which, for example, there are a plurality of hard disks aligned on a spindle and a plurality of aligned recording heads for each of the disks. One of the disks can be a reference or servo disk having only the reference track 154, with the other disks having only the data tracks T₀–T₁₅₃. The position control circuit 166 with the recording head 28 can be used exclusively with the reference disk. As the recording head 28 is moved to positions relative to the reference track 154, the other recording heads will become aligned with their corresponding data tracks T₀–T₁₅₃. This assumes all of the disks on the spindle will expand and contract, and rotate eccentrically, if at all, to the same degree. Further, the position control circuit 166 can be used in connection with disks or diskettes having data track densities other than that indicated for medium 86.

FIG. 15 illustrates a disk clamp mechanism 244 that is used to clamp a diskette 18 onto spindle 22. Disk clamp mechanism 244 includes a clamp arm 246 that is pivotally mounted on base 14 at 248 and a disk clamp 245 that extends over spindle 22. Clamp arm 246 carries a cam 250 having a slanted cam surface 252 and a straight cam surface 254 on which cam follower 148, which is connected to pick head 92, operates. This is shown in FIG. 8, can ride. A spring 256 biases clamp arm 246 and, hence, disk clamp 245, to an upper or unclamped position, as shown in FIG. 16.

FIG. 16 shows the disk clamp mechanism 244 and the cam follower 148 when pick head 92 is moving in a direction towards the operative position. Then, as pick head 92 approaches the operative position, cam follower 148 rides along cam surface 252 to bias clamp arm 246 and disk clamp 245 downwardly against the bias of spring 256. As cam follower 148 approaches the top of cam surface 252, diskette 18 becomes clamped on spindle 22. As pick head 92 continues moving beyond the operative position, cam follower 148 moves onto surface 254 to continue this disk clamping action, as shown in FIG. 17.

As previously mentioned in connection with position control circuit 166, the amplitude of waveform 160 being detected along portion 164 may not fall within the range <0.8F and > the value 5. This may be due to a relatively large misclamping of diskette 18 on spindle 22, resulting in a relatively large eccentric rotation. If this occurs, microprocessor 180, under program control, energizes motor 142 to move carriage 96, pick head 92 and cam follower 148 from the FIG. 17 position back towards a position at which cam follower 148 has partially moved down surface 252, but depending member 147 has not yet contacted surface 150 of diskette 18.

Then, microprocessor 180 energizes motor 142 to move carriage 96, pick head 92 and cam follower 148 back to the FIG. 17 position. This action, it can be appreciated, will unclamp and then reclamp diskette 18 onto spindle 22. If this results in substantially accurate clamping of the diskette 18 on spindle 22 such that the above-mentioned amplitude range is satisfied after additional calculations by microprocessor 180, then no further reclamping action is performed. Otherwise, the reclamping action continues at least several times until such an amplitude range is satisfied.

As also previously mentioned, and with reference to FIG. 8, when head 92 is moved to retrieve a diskette 18 from disk pack 42, cam follower 104 moves over and beyond incline 118 and incline 120 of cam surface 116. However, when pick head 92 is moved to return a diskette 18, cam follower 104 moves only as far as, or onto, incline 118. This control of the movement or positioning of cam follower 104 occurs in the following manner.

Carriage 96 carries the flag 149a which moves into the optical sensor 149b. This is the "home" position of carriage 96, at which cam follower 104 is on the incline 118. When retrieving a diskette 18, under program control, microprocessor 180 energizes motor 142 to drive carriage 96 to its "home" position and this is sensed by microprocessor 180 via sensor 149b. Thus, microprocessor 180 knows where carriage 96 and pick head 92 are. Then, under program control, microprocessor 180 drives head 92 further towards the retrievable position. When the head 92 is now driven with diskette 18 towards the operative position, flag 149a is removed from optical sensor 149b so that again microprocessor 180 knows the position of head 92. Thereafter, when head 92 pushes on diskette 18 to return it to disk pack 42, flag 149a again is moved into optical sensor 149b at the "home" position at which cam follower 104 is on incline 118. Microprocessor 180, therefore, again knows of the position of carriage 96 and deenergizes motor 142 to stop further movement of head 92. The previously described detent 52 then draws diskette 18 into disk pack 42.

Similarly, when retrieving a diskette 18, the box 16, under program control, is moved to its home position at which flag 41a is moved into optical sensor 41b. This "home" position is the position where one of the grooves 60, for example, groove 60-1, is in the retrievable position. Microprocessor 180 thus knows the position of box 16. Then, under program control, motor 80 can be energized by microprocessor 180 to drive box 16 to position a desired groove 60 in the retrievable position in anticipation of retrieving a selected diskette 18.
While not shown, there also can be included as part of the disk drive a similar flag and optical sensor to inform the microprocessor 180 when door 38 is opened and closed. If, for example, the door 38 is opened when a diskette 18 is in the operative position on spindle 22, microprocessor 180, under program control, can actuate picker mechanism 20 to drive diskette 18 back into disk pack 42 in the retrievable position.

Computer program listings for an 8080 microprocessor for performing the above-described functions are included as part of the specification at the end of the description and before the claims. A description of the title and general content of the listings is given below.

1. Command Input For PICPAC—This program implements the procedure for pivoting box 16 to bring a selected diskette into the retrievable position and for moving the selected diskette between the retrievable position and the operative position. It brings the above-described components into the "home" positions and controls the clamping and reclamping of the diskette to reduce or eliminate eccentric rotation.

2. Position Control Module—This program implements the procedure for compensating for expansion, contraction and eccentric rotation of the data storage medium 86 when moving the head 28, but assumes that amplitude F of level portion 162 is constant at each of the 8 circumferential locations along reference track 154.

To summarize the overall operation of disk drive 10, a disk pack 42 is loaded into tray 32 with the plurality of diskettes 18. Under software control, microprocessor 180 controls the pivoting of box 16 to move a selected one of the diskettes 18 into the retrievable position. Then, picker mechanism 20, under software control, is actuated to pick the one diskette 18 from the retrievable position, move it into the operative position, and clamp it or reclamp it on spindle 22. Then, under software control, spindle 22 is driven to rotate the data storage medium 86 and position control device 24 is actuated to seek and accurately follow the data track T0-T133, whether or not the medium 86 expands, contracts or rotates slightly eccentrically. After using the selected diskette 18, picker mechanism 20, under software control, moves the diskette 18 from the operative position to the retrievable position in anticipation of the selection of another diskette 18.

In addition to the advantages specifically described or indicated above, the picker mechanism 20 uses only a single motor 142 to move and clamp a selected diskette 18 on spindle 22. The process of picking a selected diskette 18 from the retrievable position, unclamping it from the picker mechanism 20 at the operative position, clamping the diskette 18 on spindle 22, unclamping the diskette 18 from spindle 22, and returning the diskette 18 to the retrievable position in disk pack 42 is synchronized by means of a mechanical picker mechanism 20 that requires only the one motor 142 to drive carriage 96. Furthermore, pick head 92 is not driven onto the edge 150 of diskette 18, but clamps the diskette 18 through slot 88 without causing wear or other damage to the jacket 84 and data storage medium 86. Nor does the pick head 92 hold the diskette 18 by friction.

Other aspects, objects and advantages of the invention can be obtained from a study of the drawings, the disclosure and the appended claims.

```
ORG 100H;
START: ;COMMAND INPUT FOR PICPAC
11/24/81;
;DICK
RCB EDU OAH
C R EDU 0DH
LF EDU OAH
DSK EDU OF46H
ROG EDU 700H
;NEXTCH:
MVI C,1 ;CONSOLE IN COMMAND
CALL 5H ;CALL TO BIOS
RET

CONOUT:
MVI C,2 ;CON OUTPUT COMMAND
CALL 5H ;BIOS ENTRY POINT
RET

LINEIN:
MVI C,RGB ;GET READ BUFF COMMAND
LXI D,KBDUFF ;HR=KBD BUFF
CALL 5H ;GO TO BIOS ENTRY POINT
LDA KBDUFF+2 ;GET COMMAND
CPI 'X' ;REPEAT COMMAND ?
R2 ;YES RETURN
STA SAVPTR+1 ;AND SAVE COMMAND AND
LDA KBDUFF+1 ;SAVE # OF CHARACTERS IN COMMAND
STA SAVPTR
RET

GETCH:
XRA A ;CLR A
LXI H,KBDUFF+1 ;POINT TO # OF CHARACTERS IN BUFFER
```
; Reset character buffer for repeat command.
; REPEAT:
; LDA SAVEPTR ; Get # of characters in command
; STA KBDBUF+1 ; Put it in buffer
; LDA SAVEPTR+1 ; Get previous command
; STA KBDBUF+2 ; Put it in buffer
; CALL RETEND ; Reset keyboard pointer
; ; DO COMMAND UNTIL CR KEY
; DOFE:
; MVX C+0PH
; CALL 005
; ORA A
; JZ REPEAT
; JMP GETMORE
; ; PRIMSG:
; MOV A,M ; Get char for print
; CPI 'x' ; Return on # char
; MOV E,A ; Put char in E for sending
; PUSH H ; Save string ptr
; CALL CONOUT ; Print it
; POP H ; Restore str ptr
; INX H ; Point to next char
; JMP PRIMSG
; ; START:
; LXI SP,100H ; Point to input prompt
; CALL D$SELECT ; Disable PIC
; MOVL A,PAC
; CALL D$SELECT ; Get step table ptr to 0
; LXI A+0 ; Init phase bits in port
; CALL HOMEPIC ; Get character from buffer
; CALL STORAGE ; Read string
; LOOP:
; LXI H;INPKMT ; Point to input prompt
; CALL PRIMSG ; Output string
; CALL LINELIN ; Get line from console
; NEXT:
; LXI H;NEXT
; CALL GETCH ; Get character from buffer
; ORA A ; No more characters in buffer get new com
; JZ GETMORE ; Hand from buffer
; CPI 'O' ; Disc
; CPI 'H' ; Home
; CPI 'X' ; Repeat
; CPI 'R' ; Exit
; CPI 'T' ; Set step table ptr to 0
; CPI 'Y' ; Disable PIC
; ;
4,453,188

; DISC:
CALL GET$CH
OKA A
JZ GET$MORE
LXI B,0
CPI 'O'
JZ GOOD$DISC
INR C
CPI 'I'
JZ GOOD$DISC
INR C
CPI 'J'
JZ GOOD$DISC
INR C
CPI 'K'
JZ GOOD$DISC
CPI 'L'
JZ GOOD$DISC
CPI 'M'
JZ GOOD$DISC
CPI 'N'
JZ GOOD$DISC
CPI 'O'
JZ GET$DISC
CPI 02
JZ STORE$DISC
CPI 'P'
JZ RE$CLMP
JMP BAD

HOME:
CALL GET$CH
OKA A
JZ GET$MORE
CPI 'P'
JZ HOME$PIC
CPI 'B'
JZ HOME$PAC
JMP BAD

GET$MORE:
POP H
CALL RET$END
JMP LOOP

BAD:
LXI H,8BAD$CMD
CALL PR$MSG
JMP GET$MORE


KBDPTR: DW KB$BUF+2 "KEYBOARD BUFFER POINTER"

SAVPTR: DW 0 "BUFFER FOR $ OF CHARS IN COMMAND"

BAD$CMD: DB OA,0DH,"BAD COMMAND STRING; RETYPE",OA,0DH,""

KB$BUF: DB OFFH "KEYBOARD BUFFER GOES FROM HERE"

RES$T: Jetzt OFFH "FOR 256 BYTES"

; Pac and nic routines
; 1/29/81
"Dick

; ver 1.0

PORT$A2 EQU OFNH
PORT$B1 EQU OESH
INP$1 EQU OFDH
PORT$A2 EQU OEBH
PORT$1$CONT EQU OE7H
PAC$ENBL EQU ODH
PAC$DIS EQU OCH
PIC$RATE EQU 20
LOAD$RATE EQU 7
CLMP$RATE EQU 20
STRT$UN$CLMP$RATE EQU 80
UN$CLMP$RATE EQU 20
STRT$UN$CLMP$STPS EQU 3

PAC$RATE EQU 5
STRT$PAC$RATE EQU 12
STRT$PAC$STPS EQU 5

PIC$POS EQU 0
DELTA EQU 12
PIC$HOME$POS EQU DELTA
MEMPIC$POS EQU DELTA+4
DISC$MPAC EQU DELTA-2
RELP$POS EQU DELTA+100
CLMP$POS EQU DELTA+135
NOC$CLMP$POS EQU DELTA+115
PIC$SENSOR EQU 04H
PAC$SENSOR EQU 02H
PIC EQU 0
PAC EQU 1
/,

FIXDLY EQU 04OH
FWD EQU 1
REV EQU 0
OFFSET EQU 18

;Get sensors from data buss.

;GET$SENSORS:
DI
PUSH A
AND INP1
OUT PORT$B2
IN PORT$B1
MOV B:A
POP A
OUT PORT$B2
MOV A:B
EI
 RET

;Moves picker head out past home sensor

;GO$OUT:
STA FD$REV
LDA DLY$AMT
STA TMP
MUL A$PIC$RATE
STA DLY$AMT
LXI B:40
CALL HL$LOOP
LDA TMP
STA DLY$AMT
RET

;Delay 2 msec for each count in DLY$AMT.

;DELAY:
PUSH B
LDA STRT$SLOW
ORA A
LDA STRT$DLY$AMT
JNZ DLYLP
LDA DLY$AMT
DLYLP: MUL B$FIXDLY
BL$LOOP: DCR B
; Steps selected motor one step from current position
; in D in direction set by FWD$REV.

LD A FWD$REV
ORA A
DEC D
JZ SET$TP
INC D
INC D

SET$TP:
LD L, H$TP$TB
MOV A, L$E
ANI L$HE
ADD L
MOV L, A
JNC SG
INC H

SS:
DI
IN PORT$A2
ANI L$9FH
ORA M
OUT PORT$A2
EI
CALL DELAY
RET

; Steps selected device from CUR$POS to new position in D.

CMPL$H:
MOV A, L
CMA
MOV L, A
MOV A, H
CMA
MOV H, A
RET

; Resets start delay if we have gone STRT$TPS steps.

START$DELAY:
LD L, H$STRT$TPS
LD A, STRT$CTR
INR A
STA STRT$CTR
CMP M
RZ
XRA A
STA STRT$SLOW
RET

; STEP$IT:
LHLD CUR$POS
CALL CMPL$H
ADD D
JNC STO
MOV A, FWD
INX H
JMP STI

STO:
CALL CMPL$H
MOV A, REV
STI:
STA FWD$REV
MOV B, H
MOV C, L
LHLD CUR$POS
XCHG
XRA A
STA STRT$CTR
H$LOOP:  
    MOV  A,B
    ORA A
    JZ  H$LOOP
    LDA STRT$SLOW
    ORA A
    CNZ START$DELAY
    CALL SND$STP
    DCR B
    JMP H$LOOP

L$LOOP:  
    MOV A,C
    ORA A
    LLP:  
    JZ  S$EXIT
    LDA STRT$SLOW
    ORA A
    CNZ START$DELAY
    CALL SND$STP
    DCR C
    JMP LLP

S$EXIT:  
    XCHG
    SHLD CUR$POS
    XCHG
    RET

; Selects PAC motor if B=1; PIC motor if B=0.
; SELECT:
    LDA PIC$PAC
    CMP B
    LHLD CUR$POS
    XCHG
    JZ  OK
    LHLD IMP$POS
    SHLD CUR$POS
    XCHG
    SHLD IMP$POS
    CALL SET$STP

OK:  
    MOV A,B
    STA PIC$PAC
    ORA A
    DI
    JNZ SEL$PAC
    IN  PORT#$A2
    ANI 07FH
    OUT PORT#$A2
    EI
    RET

SEL$PAC:  
    MVI A,PAC$ENBL
    OUT PORT$1$CONT
    MVI A,PAC$RATE
    STA DLY$AMT
    EI
    RET

; Disable selected motor.
; D$SELECT:
    LDA PIC$PAC
    ORA A
    DI
    JNZ D$SEL$PAC
    IN  PORT#$A2
    ORT 00H
    OUT PORT#$A2
    EI
    RET

D$SEL$PAC:  
    MVI A,PAC$DIS
    OUT PORT$1$CONT
    EI
    RET
; Select PIC motor and home it.

HOMEPIC:
MVI B::PIC
CALL SELECT
MVI A::PICRAT
STA DLY::AMT

; Home without selecting

PIC::HOME:
CALL GET::SENSORS
ANI PIC::SENSOR
MVI A::FWD
CNZ GOUT
MVI A::REV
STA FWD::REV

H::PIC:
CALL GET::SENSORS
ANI PIC::SENSOR
JZ STPR
MVI A::OFS
ANA E
JZ H::EXT

STPR:
CALL SNB::STP
JMP H::PIC

H::EXT:
LXI H::PIC::HOMEPOS
SHL D CUR::POS
XCHG CUR::POS
CALL D::SELECT
RET

; Get the disc from PAC, move it in to drive and clamp it

GET::DISC:
MVI A::PICRAT
STA DLY::AMT
MVI B::PIC
CALL SELECT
LXI D::PIC::POS
CALL STEP::IT
MVI A::LOAD::RAT
STA DLY::AMT
LXI D::REL::POS
CALL STEP::IT
MVI A::CLMP::RAT
STA DLY::AMT
LXI D::CLMP::POS
CALL STEP::IT
CALL D::SELECT
RET

; Put disc back in PAC, move picker to index PAC position

STORE::DISC:
MVI B::PIC
CALL SELECT
MVI A::OFFH
STA STRT::SLOW
MVI A::STR::UN::CLMP::RAT
STA STRT::DLY::AMT
MVI A::STR::UN::CLMP::STPS
STA STRT::STPS
MVI A::UN::CLMP::RAT
STA DLY::AMT
LXI D::UN::CLMP::POS
CALL STEP::IT
MVI A::LOAD::RAT
STA DLY::AMT
CALL PIC::HOME
MVI B::PIC
CALL SELECT
LXI D::DISC::IN::PAC
MVI A::PICRAT
STA DLY::AMT
CALL STEP::IT
MVI A::LOAD::RAT
STA DLY#AMT
LXI B:#NEWPIC#POS
CALL STEP#IT
CALL $#SELECT
RET

; reclamp the disc in the drive
RECLMP:
MVI B:#PIC
CALL SELECT
MVI A:#OFFH
STA STR#SLOW
MVI A:#STR$UN#CLMP$RATE
STA STR#BLY#AMT
MVI A:#STR$UN#CLMP$STPS
STA STR#STPS
MVI A:#UN#CLMP$RATE
STA DLY#AMT
LXI B:#UN#CLMP$POS
CALL STEP#IT
MVI A:#CLMP$RATE
STA DLY#AMT
LXI B:#CLMP$POS
CALL STEP#IT
CALL $#SELECT
RET

; Home the PAC motor
HOME#PAC:
MVI B:#PAC
CALL SELECT
MVI A:#REV
STA FWD#REV
HOME#PAC$L:
CALL GET#SENSORS
ANI PAC$SENSOR
JZ STPB
MVI A:#SH
ANI E
JZ EXTB
STPB:
CALL SET$SLOW
CALL SND#STP
JMP HOME#PAC$L
EXTB:
LXI H#P
SHLD CUR#POS
LHLD DISC$TBL
XCHG
CALL SET$SLOW
CALL STEP#IT
CALL $#SELECT
RET

; Go to disk specified in B#C
GO#DISC:
MOV A/#C
RLC
MOV C#:A
PUSH B
MVI B:#PAC
CALL SELECT
POP B
LXI H#:DISC$TBL
DAD B
MOV E#:H
INX
MOV B#:H
CALL SET$SLOW
CALL STEP#IT
CALL $#SELECT
RET

;
;SET UP FOR SLOW START OF PAC.

; SETSLOM:

MVI   A,OFFH
STA   SLOMLOW
MVI   A,SLOMPACRATE
STA   SLOMPYANT
MVI   A,SLOMPACSTPS
STA   SLOMPSTPS
RET

; DISCSTBL:

DW   OFFSET,116+OFFSET,237+OFFSET
DW   350+OFFSET,472+OFFSET

TMPPOS:

DW   0

CURPOS:

DW   0

TMP:

DB   0

STEPBL:

DB   60H,40H,0,20H

DLYYANT:

DB   2

FWDREV:

DB   FWD

PICS PAC:

DB   PIC

SLOMPSTPS:

DB   5

SLOMPCTR:

DB   0

SLOMLOW:

DB   0

SLOMPYANT:

DB   3

END

; This code implements the position control functions required in the Ailson drive. It is called from the PL/I code.

PUBLIC DELTAPOS, GOPOS, CURPOS, HOME0
PUBLIC GO$NEW$POS, THERE?, PUT$NEW$POS
EXTRN DELAY, GETBPB, PUTBPB, PUTBPC, RDDRVP, RDPOS

MAXSTEP EQU 31
STEPMASK EQU 7

; Parameters for the micro-stepping routine

DIRBIT EQU BIT1
HIBIT EQU BIT2
FSBIT EQU BIT3

; WARNING: Disk dependent parameters

TRACKSZE EQU 10
STEP$DLY EQU 300
STEP$SETTLE EQU 3225
STEP$TIME EQU 3225

SETTLE$TIME EQU 32767

; DELTAPOS is called from PL/I with a 16-bit argument specifying the number of units that the drive is to move.

DELTAPOS:

DE@HL iPick up the argument
XCHG DE@HL

DE@HL

NOT INT

LHL NEWPOS iAdd the argument to NEWPOS

DAD D

SORD NEWPOS iAnd make it the new position

RSTINT CALL GO$NEW$POS iPerform the move

RET iGOPOS moves the head to an absolute position.

GOPOS:

DE@HL iGet the argument
XCHG DE@HL

XCHG

SHLD NEWPOS iInto HAL

CALL GO$NEW$POS iMove it

RET iCURPOS returns the current position to a PL/I program.

PUBLIC GO$NEW$POS, THERE?, PUT$NEW$POS
EXTRN DELAY, GETBPB, PUTBPB, PUTBPC, RDDRVP, RDPOS

CURPOS:

CALL RDPOS iReturn POSCTR to PL/I
In case they're only looking at the lower 8 bits

RET

; DoneNewPOS is called to move the head until POSCTR equals NEWPOS.

CALL DoneNewPOS ;Move the head

LXI (1uSTEPSETTLE) ;Wait for it to do

CALL Delay

CALL DoneNewPOS ;and make sure it's there

; DoneNewPOS performs a move of the head to bring POSCTR into
; conformance with NEWPOS. It has two stages. If the distance
; to be moved is at least a full step, it will move the head to
; the nearest full step in the desired direction and then keep
; moving the head in full steps until the remaining distance is
; less than a full step. When the distance is less than a full
; step, the head will be micro-stepped in the desired direction
; until it arrives at the desired position. Note that the distance
; to be moved is re-calculated each time through one of the loops.
; This allows an interrupt routine to modify NEWPOS during execution
; of the routine with the proper response.

CALL CalcDist ;Calculate distance to be moved

RC ;Exit if the distance is zero

SAVREG H

LXI D-TRACK*SIZE+1 ;Are we at least a full step away?

JNZ DoneNewPOS

CALL USTEP ;No. Micro-step it

GOSTEP

HSTEPSETTLE ;Go to nearest full step in desired direction

LXI HSTEPSETTLE ;Let it settle

CALL DELAY

JMP DoneNewPOS

GOSTEP

HSTEPSETTLE ;Go to nearest full step in desired direction

LXI HSTEPSETTLE ;Let it settle

CALL DELAY

JMP DoneNewPOS

FSTEP: RWREG H ;Clean up the stack

CALL GOSTEP ;Go to nearest full step in desired direction

LXI HSTEPSETTLE ;Let it settle

JNC DoneNewPOS ;If a move occurred

FLOOR: CALL CalcDist ;Can we take another full step?

LXI D-TRACK*SIZE+1

JNZ DoneNewPOS

CALL Step ;No. Go micro-step

LXI HSTEPSETTLE ;Hand let it settle

CALL Delay

JMP Floor

; GOSTEP moves the head to the nearest full-step position in the
; direction indicated by DIRECT. If the head is already at a full-
; step position, it returns with the carry-bit set.

GOSTEP:

LDA CURUSTEP ;Are we there already?

CPI MAXUSTEP

STC ;Assume we are

JNZ Gostep ;Yes. Do nothing

LDA DIRECT ;Step increment is two if DIRECT

ADD ANI 2 ;is outward, and zero otherwise

MOV CXA ;Save it

LXI HCURUSTEP ;l-1 if step is odd

MOV AXM ;Get LSB of step into carry

RRA

MOV AXC ;and subtract the carry from the increment

ADD M

ANL SMASK ;Keep the step in range

MOV MXA ;Save it off

CALL DOSTEP ;Rand put it out to the hardware

MOV AXA ;MAXUSTEP ;Set CURUSTEP to the full-step position

STA CURUSTEP

MVI A;COUNTERIMODE ;Tell the hardware

OUT COUNTERCONTROL
LDA uSTEP*BL+MAXSTEP
OUT COUNTER1
ORA A ;Clear the carry bit
Gofret: RET
; CALCUST calculates the remaining distance and leaves it
; in H&L. In addition, it sets DIRECT appropriately if the distance
; is zero and returns with the carry bit set if it is zero.
; THERE? compares the current position with NEMPOS, returning with
; the carry bit clear if they are equal, and set if they aren't.
CALCUST:
CALL RDPOS ;Get current position
XCHG
LHLD NEWPOS ;Get desired position
MOV A\L
SUB E
MOV L\A
MOV A\H
SBB D
MOV \A
ORK L ;Don't change DIRECT if distance is zero
JZ Clcret ;but return with the carry bit set
MOV A\H ;Load DIRECT from sign of result
RAL
SBB A
STA DIRECT
JZ Posdst ;Form absolute value
NEGHL
Posdst: STC ;Return with the carry bit clear
Clcret: CMC
RET
THERE?: CALL RDPOS ;Load up current position and NEWPOS
XCHG
LHLD NEWPOS ;Subtract them
MOV A\L
SUB E
MOV L\A ;We only need to save the lower byte
MOV A\H
SBB D
ORK L ;A=0 if result is zero
JZ AD ;If result is zero, return with carry
MOV A\H ;Set the carry.
AD
NEG
IIf result is zero, return with carry clear
RET
uSTEP: ;micro-steps the head by one in a direction determined
; by DIRECT. (DIRECT = 0 moves the head outward; DIRECT = -1 moves it
; in.) It is table driven, and the index into the table is
; defined as follows:

; Bit 0 -- Current step is odd
; Bit 1 -- Direction is inward
; Bits 2, 3, and 4 --
; 000 -- Normal
; 001 -- CURuSTEP = 0 (Half-step if step is odd)
; 010 -- CURuSTEP = 1 (Half-step if step is even)
; 011 -- CURuSTEP = MAXuSTEP-1 (Full-step if step is odd)
; 100 -- CURuSTEP = MAXuSTEP (Full-step if step is even)

uSTEP:
LXI H\CURuSTEP ;Pick up the current micro-step
MOV E\M ;Form an index out of it
MOV D\O
MOV C\E ;Save it for later
LXI H\uSTEP*CLASS ;Get bits 2, 3, and 4
DAD D
MOV E\M ;of the table index
LDA DIRECT ;Add bit 1 from the direction
ANI D\RBIT
ORK E ;Mix it in
MOV E\A ;and save it
LDA CURuSTEP ;Now bit 0 from LSB of CURuSTEP
MOV B\A ;But save it first
ANI BITO
ORK E ;And do the mix
MOV E\A ;Back to form the index
; See if there's a step change
; There isn't
; There is. Add it to the saved step value
; Keep it in range
; Save it off
; iand put it out to the hardware
; Now add in the micro-step change
; iand save the result
; Translate from logical to physical
; But first set up the counter
; Get the physical value
; Put it out to the counter

; uSTEP$CLASS:
  DB 004H, 008H, 20H, 0
  ENDM
  DB 00CH, 010H

; uSTEP*INC:
  DB 0, 0, 0, 0  ; Neither half-step nor full-step
  DB 0, -1, 1, 0  ; CURUSTEP = 0
  DB 0, 0, 0, 0  ; CURUSTEP = 1
  DB 0, 0, 0, 1  ; CURUSTEP = MAXUSTEP-1
  DB -1, 0, 0, 1  ; CURUSTEP = MAXUSTEP

; uSTEP*INC:
  DB 1, -1, -1, 1  ; Normal
  DB 1, 1, -1, 1  ; CURUSTEP = 0
  DB 1, -1, -1, 1  ; CURUSTEP = 1
  DB 1, -1, -1, 1  ; CURUSTEP = MAXUSTEP-1
  DB -1, -1, -1, 1  ; CURUSTEP = MAXUSTEP

; uSTEP*STBL:
  DB 1, 5, 9, 13, 15, 17, 19, 21
  DB 23, 25, 27, 29, 31, 33, 35, 37
  DB 39, 42, 45, 48, 51, 54, 57, 61
  DB 65, 69, 73, 77, 82, 87, 93, 100

; STEP moves the head by one full step in the direction determined
; by DIRECT. Note that it is the responsibility of the calling
; routine to wait an appropriate length of time before trying
; to move the head again.

; STEP:
  LDA 0
  ORN A
  JP Strout
  MV: 1
  E, 2  ; Assume DIRECT = 0 and set increment
  LDA 0
  ORN A
  JP Strout
  MV: 1
  E, 2  ; Right. Leave it alone
  MV: 1
  E, 2  ; Wrong. Change it
  LDA CURUSTEP
  STA CURUSTEP
  CALL DOSTEP
  RET

; DOSTEP takes a logical step position in the accumulator,
; translates it to a physical value, and puts it out to the
; hardware.

; DOSTEP:
  MOV E, A  ; Make of the value a table index
  MX: 0, 0
  LDA H, STEP*STBL
  PERFORM the translation
  DAD 0
  MOV A, M
  CALL PUTBPC
  RET
HOMED: puts the head into the position from which to start, looking for the outer servo track. It first moves the head outward to the nearest full-step position. It then steps outward in full steps until the drive track zero sensor is true and CURSTEP is 0.

HOMED:

XRA A ; Set step direction
STA DIRECT ; To outward
CALL GOSTEP ; Go to the nearest full step
LXI H;STEP&TIME ; And let it settle
CNC DELAY ; If we moved

Fullst#:

CALL ROOKV ; Track zero sensor set?
ANI TRACKO
JNZ Gostep ; No. Go move the head
LDA CURSTEP ; Yes. Are we at step 0?
ORA A
JZ Hdhome ; Yes. We're done

Gostep:

CALL Stellar ; Not yet. Step out and try again
LXI H;STEP&TIME ; Wait for it to settle
CALL DELAY
JMP Fullste

Hdhome:

LXI H;SET&TIME ; Let it quiet down
CALL DELAY
CALL GETBPB ; Clear the position counter
SARVREG PSW
ORI CLPOS
CALL PUTBPB
RSTREG PSW
CALL PUTBPB
LXI H=0 ; and NEMPOS
SHLD NEMPOS
RET

DOES:

NEMPOS: DW 0 ; Position to which to move
DIRECT: DW 0 ; Direction of desired move
IF = outward, -1 = inward.
CURSTEP:

DB 0 ; Current step
CURuSTEP:

DB MAXSTEP ; Current micro-step

END

; This module contains the code having to do with index pulses and segment interrupts. It contains the segment interrupt handler and the routines which find, start, interrupt, and stop index pulses.

PUBLIC SEGINT, CURSEG, STOPBP, STRIP, INTIP, SYNCIP
EXTRN SEGFSH, GETBPB, PUTBPB, STRTP, STOPCP

; States. Used as indexes into a table, so increment by two.
SYNCFIND EQU 0
INDEX EQU 2
SEGMENT EQU 4
SEGMASK EQU OFH ; Mask for valid segment numbers
INDEXTIME EQU 8400 ; Timer count to separate index pulses
SEGINT is the segment interrupt code. It maintains CURRENTSEG; calls the Segment Finite State Machine if appropriate, and controls the passing through of the index pulses to the controller.

SEGINT:

EI
SAVREG (PSW+B:10H)
LXI D;STATE&VEC ; Go to the appropriate routine
LHLD STATE
DAD D
DECHL
XCHG
CALL@HL
RSTREG (H=0:B:PSW)
RET

STATE&VEC:

DW GTSYNC ; SYNCFIND
GTSYNC is the routine for the interrupt handler in SYNCSFIND state.

GTSYNC:

CALL STOPCK
; Get the time from the last pulse
JC Notsyn
; If the clock hasn't been started yet
LXI A-INDEXTIME
; Check the interval
DAU D
; It was long. This isn't it
XRA A
; Eureka. Set CURRENT$SEGMENT to zero
LXI H$SEGMENT
; and the state to normal
SHLD STATE
JMP Gtsret
; Finish up and leave

Notsyn:
CALL STRICK
; Reset the timer and start it up again

Gtsret:
SNEDEI RET
; DOIP handles the interrupt when this is an index pulse.

DOIP:
MOINT
; Gate the pulses off from the controller
ANI NOT DRVSTATEA
CALL PUTBPA
RSTINT
LXI H$SEGMENT
; Reset the state
SHLD STATE
SNEDEI RET
; DONOTIP handles the interrupt when this is a normal segment interrupt.

DONOTIP:
LDA CURRENT$SEGMENT
; Advance the segment counter
INR A
ANI SEGMASK
; Keep it in range
STA CURRENT$SEGMENT
CZ INDEXNEXT
; If an index pulse is coming up
SNEDEI LDA CURRENT$SEGMENT
; Allow the segment routine to be interrupted
CZ SEGFSM
; If the current segment is even
ANI ГITO
; Process it in the PL/I code
RET
; INDEXNEXT examines IPFLAG to see what to do about the upcoming index pulse. IPFLAG = -1 causes the index pulse to be passed through to the controller. IPFLAG = 1 causes the pulse to be locked out.
; IPFLAG = 0 causes this one to be skipped but succeeding ones to be let through.

INDEXNEXT:
LXI H:INDEX
; Change the state
SHLD STATE
LDA IPFLAG
; See what we're supposed to do
OKA A
JM Indxon
; -1. Pass it on
JMZ Inret
; 0. Leave it locked out
DCR A
; J0. Ignore this one and reset the flag
STA IPFLAG
JMP Inret

Indxon:
MOINT
CALL GETBPA
; Enable the index pulse back to the controller
ORI DRVSTATEA
CALL PUTBPA
RSTINT
Inret: RET
; These routines are all PL/I callable. CURSEG returns the current segment number. STOPIP, STARTIP, INITIP, and SYNCP Stop start, interrupt, and find the index pulses.

CURSEG:
LDA CURRENT$SEGMENT
OKA A
; Divide it by two for PL/I
RAR
MUV L:A
; PL/I wants it in both A and HL
MVI H:0
RET

STOPIP:
MVI A/1
; Disable index pulses until
STA IPFLAG
; STARTIP is called
RET
STRTIP:
MVI A,-1
STA IPFLAG
RET
INTIP:
LDA IPFLAG
CPI -1
JNZ Inv Id
XRA A
LYR:
STA IPFLAG
Inv Id:
RET
SYNIP:
LXI HL,SYNC$FIND
SET the state to find the index
SHLD STATE
Wsync:
LHLD STATE
and wait until it's been found
MOV A,1
CPI LOW SYNC$FIND
JZ Wsync
RET
DSEG STATE:
DB sync$FIND
STATE variable for interrupt handler
CURRENT$SEGMENT:
DB 0
IPFLAG: DB 1
Index pulse lock flag
J= Lock
0 = Lock and enable next one
J-1 = Enable
END

Segment Finite State Machine 000.10 3/24/81
This file contains the state machine activated when there is a segment clock interrupt. It is extremely simple.

The machine has four states.
In the IDLE state, it does nothing.
In the SERVO_READ state, it records one revolution's worth of servo information and returns to the IDLE state.
In the TRACK_FOLLOW state, it pulls a correction value out of the CORRECTION table and, if it is different from the current one, moves the head by the difference.
In the END_TRACK_FOLLOW state, it moves the head back to the position it was in when track-following started, and sets the state to IDLE.

/*
** Segment Finite State Machine
**
PROCEDURE SEGMENT_FINITE_STATE_MACHINE:
DECLARE
  ENTRY VARIABLE STATIC:
STATE, END_SEGMENT,
CURRENT_CORRECTION,
TBPTR POINTER,
CORRECTION (0..MAX_SEGMENT)
 ) STATIC,
STATE_VECTOR (4) ENTRY VARIABLE STATIC;
DECLARE
CURSEG ENTRY RETURNS (FIXED_BINARY),
SELSEG ENTRY (FIXED_BINARY),
DELETAPOS ENTRY (FIXED_BINARY);
INSFSM: /* Initialize the Segment FSM */
PROCEDURE EXTERNAL;
STATE_VECTOR(IDLE) = DO_IDLE;
STATE_VECTOR(SERVO_READ) = DO_SERVO_READ;
STATE_VECTOR(TRACK_FOLLOW) = DO_TRACK_FOLLOW;
STATE_VECTOR(END_TRACK_FOLLOW) = DO_END_TRACK_FOLLOW;
STATE = IDLE;
END INSFSM;
SEGFSM: /* Segment Finite State Machine */
PROCEDURE EXTERNAL;
CALL STATE_VECTOR(State)();
END SEGFSM;

GET SqlDataReader:
/* Read a track of servo data */
PROCEDURE (TRACK_DATA) EXTERNAL;
DECLARE
TRACK_DATA (0:MAX_SEGMENT) FIXED;
TDPTR = ADDR(TRACK_DATA);
CALL SELSRV(OUTER);
STATE = SERVO_READY;
END_SEGMENT = CURSEG();
DO WHILE (STATE = SERVO_READ);
END;
END GET SqlDataReader;

SET_CORRECTION:
PROCEDURE (NEW_CORRECTIONS) EXTERNAL;
DECLARE
NEW_CORRECTIONS (0:MAX_SEGMENT) FIXED;
DECLARE
SEG FIXED (7);
DO SEG = 0 TO MAX_SEGMENT;
CORRECTION(SEG) = NEW_CORRECTIONS(SEG);
END;
END SET_CORRECTION;

STARTTRACKFOLLOW:
PROCEDURE EXTERNAL;
DECLARE
SEG FIXED (7);
CURRENT_CORRECTION = 0;
STATE = TRACK_FOLLOW;
SEG = CURSEG();
DO WHILE (SEG = CURSEG());
END;
END STARTTRACKFOLLOW;

STOPTRACKFOLLOW:
PROCEDURE EXTERNAL;
IF STATE = TRACK_FOLLOW THEN
DO;
STATE = END_TRACK_FOLLOW;
DO WHILE (STATE = END_TRACK_FOLLOW);
END;
END;
END STOPTRACKFOLLOW;

SET_SEGFSM_IDLE:
PROCEDURE EXTERNAL;
STATE = IDLE;
END SET_SEGFSM_IDLE;

DO_IDLE:
PROCEDURE;
END DO_IDLE;

DO_SERVO_READ:
PROCEDURE;
DECLARE
TRACK_DATA (0:MAX_SEGMENT) FIXED BASED (TDPTR);
TRACK_DATA(CURSEG()) = GETADP();
IF CURSEG() = END_SEGMENT THEN
STATE = IDLE;
END DO_SERVO_READ;

DO_TRACK_FOLLOW:
PROCEDURE;
DECLARE
(NEW_CORRECTION, DISTANCE) FIXED;
NEW_CORRECTION = CORRECTION(CURSEG());
DISTANCE = NEW_CORRECTION - CURRENT_CORRECTION;
IF DISTANCE <= 0 THEN
DO;
CALL DELTAPOS(DISTANCE);
CURRENT_CORRECTION = NEW_CORRECTION;
END;
END DO_TRACK_FOLLOW;

DO_END_TRACK_FOLLOW:
PROCEDURE;
CALL DELTAPOS(-CURRENT_CORRECTION);
STATE = IDLE;
END DO_END_TRACK_FOLLOW;
Servo Reading and Interpreting code, Vol. 07 DML 3/24/81
This file contains all the code to do with the track-following servo except for the code executed upon segment interrupts which is in the segment finite state machine (SEGFSM).

Servo_Module:

PROCEDURE;
#include 'DEFINES.PLI';
#include 'DSKDEF.PLI';

/* Global data */

DECLARE
(  TRACK_DATA (0:MAX_SEGMENT) FIXED;
  SERVO_DATA (0:MAX_SEGMENT) FLOAT;
  CORRECTIONS (0:MAX_SEGMENT) FIXED;
  TRACK_OFFSET, UPPER_LIMIT, INNER_SERVO_POS, OUTER_SERVO_POS) FIXED;
  CALIBRATION FLOAT;
  DC_OFFSET FIXED INITIAL (0) ) STATIC;

/* External routines */

DECLARE
GPOSC ENTRY (FIXED);
SET_CORRECTION ENTRY (0:MAX_SEGMENT) FIXED;
HOMEH ENTRY;
CURPOS ENTRY RETURNS (FIXED);
DELTAPOS ENTRY (FIXED);
PLIDLY ENTRY (FIXED);
SETP0S ENTRY (FIXED);
SELSRV ENTRY (FIXED);
GETAD ENTRY RETURNS (FIXED);
GTSRV ENTRY (0:MAX_SEGMENT) FIXED;

/* SERVOINITIALIZE is the power-up routine for this module. It makes sure that an INITIAL_SERVO.READ will be performed the first time a disk is loaded. */

SERVOINITIALIZE:

PROCEDURE EXTERNAL;
  OUTER_SERVO_POS = 0;
END SERVOINITIALIZE;

/* DOSERVOREAD is the routine called from outside to find the servo track and set up the CORRECTIONS table used for mis-clamp correction by the segment fsm. */

DOSERVOREAD:

PROCEDURE EXTERNAL;
  IF OUTER_SERVO_POS = 0 THEN
    CALL INITIAL_SERVO.READ;
  ELSE IF ≠NORMAL_SERVO.READ() THEN
    CALL INITIAL_SERVO.READ();
  END DOSERVOREAD;

GETOUTERSERVOPOS simply returns the outer servo position to an external routine.

GETOUTERSERVOPOS:

PROCEDURE EXTERNAL RETURNS (FIXED);
  RETURN (OUTER_SERVO_POS);
END GETOUTERSERVOPOS;

/* NORMAL_SERVO.READ is the routine normally used when a disk is loaded. It assumes that it will be able to read the outer servo track at the same place it did last time. If it can't, it returns a failure indication. If it can, it reads the track and re-calculates the mis-clamp CORRECTIONS table. */

NORMAL_SERVO.READ:

PROCEDURE RETURNS (BIT(1));
DECLARE
(TOO_FAR_IN, TOO_FAR_OUT) BIT(1);
CALL GOPOS(OUTER_SERVO_POS);
CALL PLIDLY(SETTE TIME);
CALL GET_SERVO(TOO_FAR_IN, TOO_FAR_OUT);
IF TOO_FAR_IN TO FAR_OUT THEN
RETURN (FALSE);
CALL GET_CORRECTION();
CALL SET_CORRECTION(CORRECTIONS);
RETURN (TRUE);
END NORMAL_SERVO_READ;

INITIAL_SERVO_READ is the routine which performs a full-enchilada servo read. It is invoked at start-up and at any other time when the normal algorithm fails or the controller requests a recalibration. It begins by calibrating the outer channel and recording its gain. (It assumes that the servo dc offset has been read before the spindle was turned on at power-up.) It then reads both the outer and inner servo tracks, records their positions, calculates the growth factor to compensate for thermal and hygroscopic growth of the disk, and calculates the current clamping error correction.

PROCEDURE INITIAL_SERVO_READ;
DECLARE FOUND_FLAG BIT(1);
(TOO_FAR_IN, TOO_FAR_OUT) BIT(1);
CALL HOMEOO();
CALL PLIDLY(SEITLE_TIME);
CALL CALIBRATE_SERVO(CALIBRATION, FOUND_FLAG);
CALL GET_SERVO(TOO_FAR_IN, TOO_FAR_OUT);
CALL GET_CORRECTION();
CALL SET_CORRECTION(CORRECTIONS);
OUTER_SERVO_POS = CURPOS() + TRACK OFFSET;
END INITIAL_SERVO_READ;

CALIBRATE SERVO is entered with the head presumed to be just outside one of the servo tracks. It makes a sweep inward to try to locate a valid servo track and if it finds one, sets CALIBRATION to the maximum value found on the outer portion of it. It also returns a success or failure indication.

PROCEDURE (CALIBRATION, FOUND_FLAG);
DECLARE
CALIBRATION FLOAT;
FOUND_FLAG BIT(1);
DECLARE
(I, SEG) FIXED(7),
(MAX_LEVEL, SAMPLE, MAX THIS TRY) FIXED,
AT EDGE BIT(1);
MAX_LEVEL = 0;
CALIBRATION = 0.000;
FOUND_FLAG = FALSE;
DO I = 1 TO MAX TRIES WHILE (~FOUND FLAG);
MAX THIS TRY = 0;
CALL GTSROO(TRACK_DATA);
DO SEG = 0 TO MAX_SEGMENT;
SAMPLE = TRACK_DATA(SEG) DC OFFSET;
IF SAMPLE > CAL TRIVIAL THEN
MAX THIS TRY = MAX(MAX THIS TRY, SAMPLE);
END;
MAX LEVEL = MAX(MAX LEVEL, MAX THIS TRY);
IF MAX LEVEL > MAX THIS TRY + HYS THERESIS THEN
FOUND_FLAG = TRUE;
ELSE
CALL DELTAPQSTRY DELTA);
ENDIF
RETURN;
UPPER LIMIT = (MAX LEVEL * 8 +5)/10;
AT_EDGE = FALSE;
DO WHILE (~AT_EDGE);
   AT_EDGE = TRUE;
   CALL GTSRV0(TRACK_DATA);
   DO SEG = 0 TO MAX_SEGMENT;
      SAMPLE = TRACK_DATA(SEG)-DC_OFFSET;
      IF (SAMPLE (# TRIVIAL) | (SAMPLE > UPPER_LIMIT) THEN
         AT EDGE = FALSE;
   END;
   IF ~AT EDGE THEN
      CALL DELTAPOS(TRACK_DATA);
   END;
   CALIBRATION = MAX_LEVEL;
END CALIBRATE SERVO;

GET SERVO attempts to read a servo track at the current head location. It checks for success (the presence of valid values all the way around the track) and, if successful, returns the outer levels divided by CALIBRATION.

*/
GET SERVO:
PROCEDURE (TOO_FAR_IN, TOO_FAR_OUT);
DECLARE (TOO_FAR_IN, TOO_FAR_OUT) BIT(1);
DECLARE SAMPLE FIXED;
SEG FIXED(7);
TOO_FAR_IN = FALSE;
TOO_FAR_OUT = FALSE;
CALL GTSRV0(TRACK_DATA);
DO SEG = 0 TO MAX_SEGMENT;
   SAMPLE = TRACK_DATA(SEG)-DC_OFFSET;
   TRACK_DATA(SEG) = SAMPLE;
   IF SAMPLE (# TRIVIAL THEN
      TOO_FAR_IN = TRUE;
   IF SAMPLE > UPPER LIMIT THEN
      TOO_FAR_OUT = TRUE;
   END;
   IF ~TOO_FAR_IN & ~TOO_FAR_OUT THEN
      DO SEG = 0 TO MAX_SEGMENT;
         SERVO_DATA(SEG) = FLOAT(TRACK_DATA(SEG))/CALIBRATION;
      END;
   END;
END GET SERVO;

GET CORRECTION transforms the ratios in SERVO DATA into the mis-clamp correction values in CORRECTIONS. At the same time, it calculates the amount by which the center of the servo track is offset from the position of the head when the track was read, and stores this in TRACK OFFSET.

*/
GET CORRECTION:
PROCEDURE;
DECLARE;
NSEGS FLOAT;
SEG FIXED(7);
DO SEG = 0 TO MAX_SEGMENT;
   CORRECTIONS(SEG) = DSTCVT(SERVO DATA(SEG));
END;
TRACK OFFSET = 0;
DO SEG = 0 TO MAX SEGMENT;
   TRACK OFFSET = TRACK OFFSET+CORRECTIONS(SEG);
END;
NSEGS = MAX_SEGMENT+1;
TRACK OFFSET = ROUND(FLOAT(TRACK OFFSET)/NSEGS-0);
DO SEG = 0 TO MAX SEGMENT;
   CORRECTIONS(SEG) = -(CORRECTIONS(SEG)-TRACK OFFSET);
END;
END GET CORRECTION;

*/
GET DC OFFSET is used at start-up to calibrate the DC offset of the servo electronics. It does this by simply reading the A/D on the assumption that there is no disk in place at the time.

*/
GET DC OFFSET:
PROCEDURE EXTERNAL;
What is claimed is:

1. Apparatus for moving a data storage device between a retrievable position at which the data storage device is stored and an operative position at which the data storage device is used, comprising:

   (a) a pick head having a pair of openable and closeable jaws;
   (b) means for carrying said pick head between the retrievable position and the operative position;
   (c) first pick cam means for opening said jaws during movement of said pick head towards the retrievable position and for then closing said jaws about the data storage device at the retrievable position, said pick head being movable with said jaws closed from the retrievable position towards the operative position;
   (d) second release cam means for opening said jaws during movement of said pick head at the operative position and for then closing said jaws beyond the operative position, said pick head being movable with said jaws closed from beyond the operative position towards the retrievable position; and
   (e) means for driving said carriage means between the retrievable position and the operative position.

2. Apparatus according to claim 1 wherein said first pick cam means comprises:

   (a) a pivotal member;
   (b) a cam member connected at a slant to said pivotal member and having an upper cam surface and an underside; and
   (c) means, coupled to said detecting means, for generating first data representing said constant amplitude F and said distance off-track in PIPS.

3. Apparatus according to claim 1 wherein said second release cam means comprises:

   (a) a pivotal member;
   (b) a cam member connected at a slant to said pivotal member and having an upper cam surface and an underside; and
   (c) means, coupled to said detecting means, for generating first data representing said constant amplitude F and a linearly sloping amplitude portion having a reference position and lead.

4. Apparatus according to claim 1 wherein said means for driving comprises:

   (a) a drive motor; and
   (b) programmable processor means for controlling said drive motor.

5. Apparatus according to claim 4 wherein said programmable processor means controls said drive motor to drive said pick head into the retrievable position beyond said first pick cam means to pick the data storage device and to drive said pick head no further than said first pick cam means to return the data storage device to the retrievable position.

6. Apparatus according to claim 1 wherein said carriage means further comprises means for controllably securing the data storage device in the operative position.

7. Apparatus according to claim 1 further comprising means for stripping the data storage device from said jaws as said jaws are opened at the operative position.

8. Apparatus, according to claim 1, wherein the data storage device has a slot, and wherein said jaws close through the slot at the retrievable position and during movement of the data storage device from the retrievable position towards the operative position.

9. Apparatus for providing servo information, comprising:

   (a) a data storage medium having a reference track having an outer diameter and an inner diameter, said reference track storing a signal having an amplitude waveform across said reference track for providing a generally level central portion of constant amplitude F and a linearly sloping amplitude portion having a reference position and leading from said generally level central portion of constant amplitude to said inner diameter of said reference track;
   (b) means, movable across said reference track, for detecting said signal to output said generally level central portion of constant amplitude F and said linearly sloping amplitude portion;
   (c) means, coupled to said detecting means, for generating first data representing said constant ampli-
tude F and second data representing the amplitude of said signal at a point on said linearly sloping amplitude portion; and
(d) means, coupled to said generating means, for determining the ratio of said second data to said first data and for determining an offset of said detecting means from said reference position \( R_{RP} \) in response to said ratio.

10. Apparatus, according to claim 9, wherein said reference position \( R_{RP} \) is at a predetermined ratio of an amplitude of said linearly sloping amplitude portion to said constant amplitude F.

11. Apparatus, according to claim 10, wherein said predetermined ratio is 0.5F.

12. Apparatus, according to claim 9, wherein said means for detecting comprises:
(a) a signal detector;
(b) stepper motor means for moving said signal detector across said reference track; and
(c) means for controlling said stepper motor means to move said signal detector in full steps to output said generally level central portion of constant amplitude F and in microsteps to output said linearly sloping amplitude portion.

13. Apparatus, according to claim 9, wherein said signal is a single frequency signal.

14. A disk drive for accessing data, comprising:
(a) a disk-type data storage medium having a plurality of concentric data tracks having centerlines and a single concentric servo track having an outer diameter and an inner diameter, said servo track storing a frequency signal having an amplitude waveform across said servo track for providing a generally level central portion of constant amplitude F and a linearly sloping amplitude portion having a reference position \( R_{RP} \) leading from said generally level central portion of constant amplitude F to said inner diameter of said servo track;
(b) first means, movable across said servo track and said data tracks, for detecting said frequency signal to output said generally level central portion of constant amplitude F and said linearly sloping amplitude portion, said first detecting means including a scale having scale lines representing the distance between said centerlines of said data tracks and the distance between said servo track and one of said data tracks adjacent said servo track, said reference position \( R_{RP} \) being a certain distance from said one adjacent data track;
(c) second means for detecting said scale lines in response to movement of said first detecting means across said servo track and said data tracks and for storing counts corresponding to the positions of said servo track and said data tracks;
(d) means, coupled to said first and second detecting means, for determining a ratio of the amplitude of said frequency signal at a point on said linearly sloping amplitude portion to said constant amplitude portion F and for determining an offset of said first detecting means from said reference position \( R_{RP} \) in response to said ratio and for determining a number representing the distance from said offset to said adjacent one data track; and
(e) means for moving said first detecting means to any one of said data tracks in dependence on said number and said counts.

15. A disk drive according to claim 14, wherein said disk-type data storage medium is expandable and contractable and said scale is expandable and contractable in proportion to said disk-type data storage medium.

16. Apparatus for storing and retrieving a data storage device, comprising:
(a) means for providing an operative position for the data storage device;
(b) means for storing the data storage device at a retrievable position;
(c) pivotal means for clamping the data storage device at the operative position; and
(d) means for moving the data storage device from the retrievable position to the operative position, including
(i) means for pivoting said pivotal means to clamp the data storage device;
(ii) a movable carriage carrying said pivotal means;
(iii) a motor for moving said movable carriage;
and
(iv) programmable microprocessor means for controlling said motor to move said movable carriage to clamp and reclamp the data storage device in response to misclamping of the data storage device at the operative position.

17. Apparatus according to claim 16, wherein said pivotal means has a pivotal clamp arm and a cam being pivotal with said clamp arm, and said pivoting means comprises a cam follower being movable along said cam.

18. Apparatus for storing and retrieving a plurality of data storage devices, comprising:
(a) means for providing an operative position for one of the data storage devices;
(b) means for storing the plurality of data storage devices and being pivotal in relation to said providing means;
(c) means for selectively pivoting said storing means to move a selected one of the data storage devices into a retrievable position, including controllable means for driving said storing means, and programmable processor means for controlling said controllable driving means; and
(d) means for moving the selected one of the data storage devices between the retrievable position and the operative position.

19. Apparatus according to claim 18 wherein said controllable driving means comprises:
(a) an index cam coupled to said storing means;
(b) a cam follower coupled to said index cam; and
(c) stepper motor means for moving said cam follower along said index cam.

20. Apparatus for storing and retrieving a plurality of data storage devices, comprising:
(a) means for providing an operative position for one of the data storage devices;
(b) means for storing the plurality of data storage devices and being pivotal in relation to said providing means;
(c) means for selectively pivoting said storing means to move a selected one of the data storage devices into a retrievable position; and
(d) means for moving the selected one of the data storage devices between the retrievable position and the operative position, including picker means for releasably clamping the selected one of the data storage devices in the retrievable position and for releasably unclamping the selected one of the data...
storage devices in the operative position.

21. Apparatus according to claim 20 wherein said picker means for releasably clamping and unclamping comprises:

(a) a pick head having a pair of openable and closeable jaws for picking the selected one of the data storage devices, and having a cam follower;
(b) carriage means for carrying said pick head between the retrievable position and the operative position;
(c) a first pick cam having a pick cam surface for opening and closing said jaws, said cam follower following said pick cam surface near the retrievable position to open said jaws about the selected one of the data storage devices and then to close said jaws and pick the selected one of the data storage devices in the retrievable position, said cam follower bypassing said pick cam surface after picking the selected one of the data storage devices and while moving towards the operative position;
(d) a second release cam having a release cam surface for opening and closing said jaws, said cam follower following said release cam surface near the operative position to open said jaws and release the selected one of the data storage devices in the operative position and then to close said jaws, said cam follower bypassing said release cam surface to push the selected one of the data storage devices with said jaws closed from the operative position towards the retrievable position; and
(e) means for driving said carriage means.

22. Apparatus according to claim 21 wherein said picker means for releasably clamping and unclamping further comprises a stripping device for stripping the selected one of the data storage devices from said jaws in the operative position as said jaws are opened.

23. Apparatus according to claim 21 wherein said carriage means comprises means for releasably securing the selected one of the data storage devices in the operative position.

24. Apparatus according to claim 20 wherein said means for storing comprises a pack having a groove and a detent biased into said groove to hold the plurality of data storage devices in said pack and to force the selected one of the data storage devices into the retrievable position in response to said picker means for releasably clamping and unclamping being near the retrievable position.

25. Apparatus according to claim 20 wherein said means for storing comprises a pack having a cut-out, said picker means for releasably clamping and unclamping being movable into said cut-out to pick and retrieve the selected one of the data storage devices.

* * * * *