

DICK BAILEY

**digital**

IMBEDDED SERVO

**RL 01**



## Sector Format

## SECTOR FORMAT DESCRIPTION

The following description is from Section One of the RL Disk Subsystem User's Guide with additions/corrections.

## RL01K CARTRIDGE FORMAT

The RL01K is a top-loading, single-disk cartridge with a total storage capacity of approximately 2.62 million 16-bit data words.

As Figure 1 shows, each recording surface has a total of 256 tracks. Combining recording surface tracks yields 256 cylinders, labeled from 0 to 255 (decimal).

Every track on a recording surface is subdivided into 40 equal length sectors, which are further subdivided into fields. The six fields in each sector contain a total of 140 words of 16 bits each, as illustrated in Figure 2. Note that only 128 of the 140 words contain data.

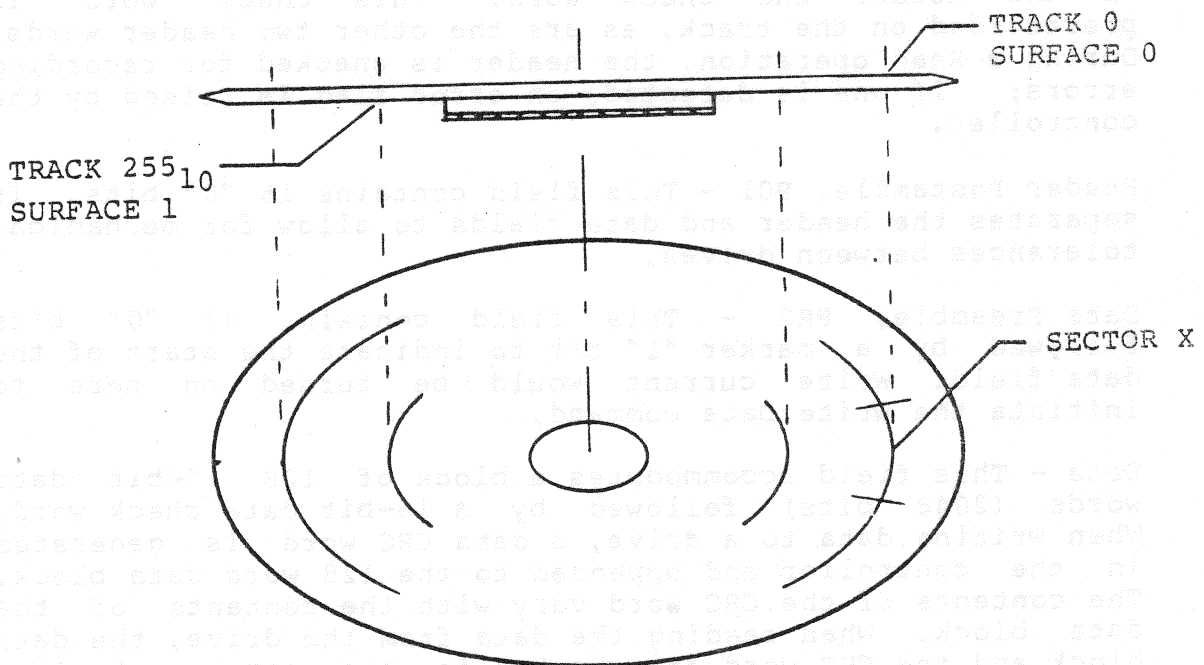


Figure 1. RL Platter

When the disk cartridge is formatted at the factory, both servo and header information are prerecorded in each sector. The servo information is contained in two pulse bursts that occur during the sector pulse. It identifies the radial position of the heads relative to each of the tracks on the cartridge and is used by the drive for track counting and head positioning. Neither servo nor header information can be modified or rewritten in the field. Accidental change is prevented by drive protection circuitry.

Contents of the six fields in a given sector include the following:

Header Preamble, PR1 - These three words precede the header information and contain 47 "0" bits followed by a marker "1" bit to indicate the start of valid header information.

Header - This field contains three words of 16 bits each. The first word identifies the drive head (upper or lower), the cylinder address (1 of 256), and the sector address (1 of 40). The second header word is all 0s. The third word is the header CRC check word. This check word is prerecorded on the track, as are the other two header words. During a Read operation, the header is checked for recording errors; if one is detected, an error flag is raised by the controller.

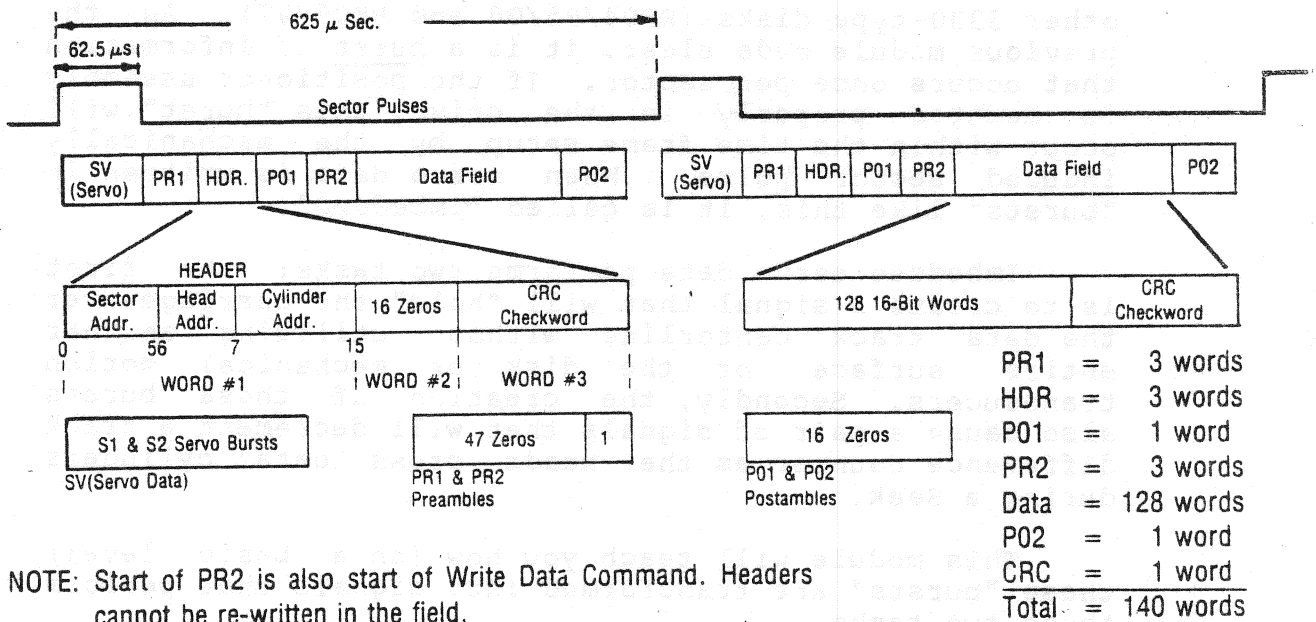
Header Postamble, P01 - This field contains 16 "0" bits. It separates the header and data fields to allow for mechanical tolerances between drives.

Data Preamble, PR2 - This field contains 47 "0" bits followed by a marker "1" bit to indicate the start of the data field. Write current would be turned on here to initiate the Write Data command.

Data - This field accommodates a block of 128 16-bit data words (2048 bits) followed by a 16-bit data check word. When writing data to a drive, a data CRC word is generated in the controller and appended to the 128 word data block. The contents of the CRC word vary with the contents of the data block. When reading the data from the drive, the data block and the CRC word are checked in the same controller circuit. Detection of a data recording error results in a data CRC flag.

Data Postamble, P02 - This field consists of 16 "0" bits. Write current is turned off at the end of this field so that data CRC information will not be disturbed.

To ensure the sector boundary required by mechanical tolerances, fixed time delays are introduced between the previous sector's P02 and the current sector's servo data field.



NOTE: Start of PR2 is also start of Write Data Command. Headers cannot be re-written in the field.

Figure 2. Sector Format

### SUMMARY

This completes the section on the cartridge format. Its purpose is to show you the similarities between this drive's format and any other disk with which you may be familiar.

# Imbedded Servo Introduction

## INTRODUCTION

The servo data used for track following in this device does not consist of a continuous stream as with other 3330-type disks (RP04/05/06 and RK06/07). As the previous module made clear, it is a burst of information that occurs once per sector. If the positioner assembly is mounted properly in the drive, this "burst" will occur within the time frame setup by the mechanically induced sector pulse. When servo data is written in "bursts" like this, it is called "imbedded".

Imbedded servo data performs two tasks: the first is to create a signal that will "hold" the carriage over the data track centerline without utilizing another entire surface of the disk or mechanical motion transducers. Secondly, the creation of these bursts also cause a pair of signals that will decrement a track difference counter as the heads cross data cylinders during a Seek.

This module will teach you how (on a basic level) these "bursts" are transformed into signals that perform these two tasks.

## Imbedded Servo Introduction

### DESCRIPTION

#### What Does the Servo Data Do for Us?

The servo data bursts are responsible to the track-counting logic for updating the code that establishes how many tracks were crossed during the time frame since the bursts were last seen (sector pulse time). Since sector pulses are 624 microseconds apart, up to three tracks could have been crossed since the last pair of "bursts" were seen. When the track counter reaches zero, the logic puts the servo system into "position" mode and allows the analog servo data to "hold" the carriage over the present track centerline.

#### What Does It Look Like?

In Figure 1 notice that under each sector pulse are two columns of waveforms. In the first "column" are the S1 bursts. They consist of two components which are given the mnemonics S1 and  $\overline{S1}$  and are identified by their opposite polarities. The second "column" of waveforms are the S2 bursts, and, like S1, they also contain the components S2 and  $\overline{S2}$ .

In the illustration, note that there are three tracks of information at the top contained within the sector pulses. These three tracks actually represent 24 tracks of consecutive  $\overline{S1}$  servo data bursts.  $\overline{S1}$  servo data has more of the signal positive than negative.

The Outer Guard Band is the period of time during which only  $\overline{S1}$  bursts are detected. This .2 inch space on the disk is officially the head-loading zone.

# RL01 SERVO DATA FORMAT

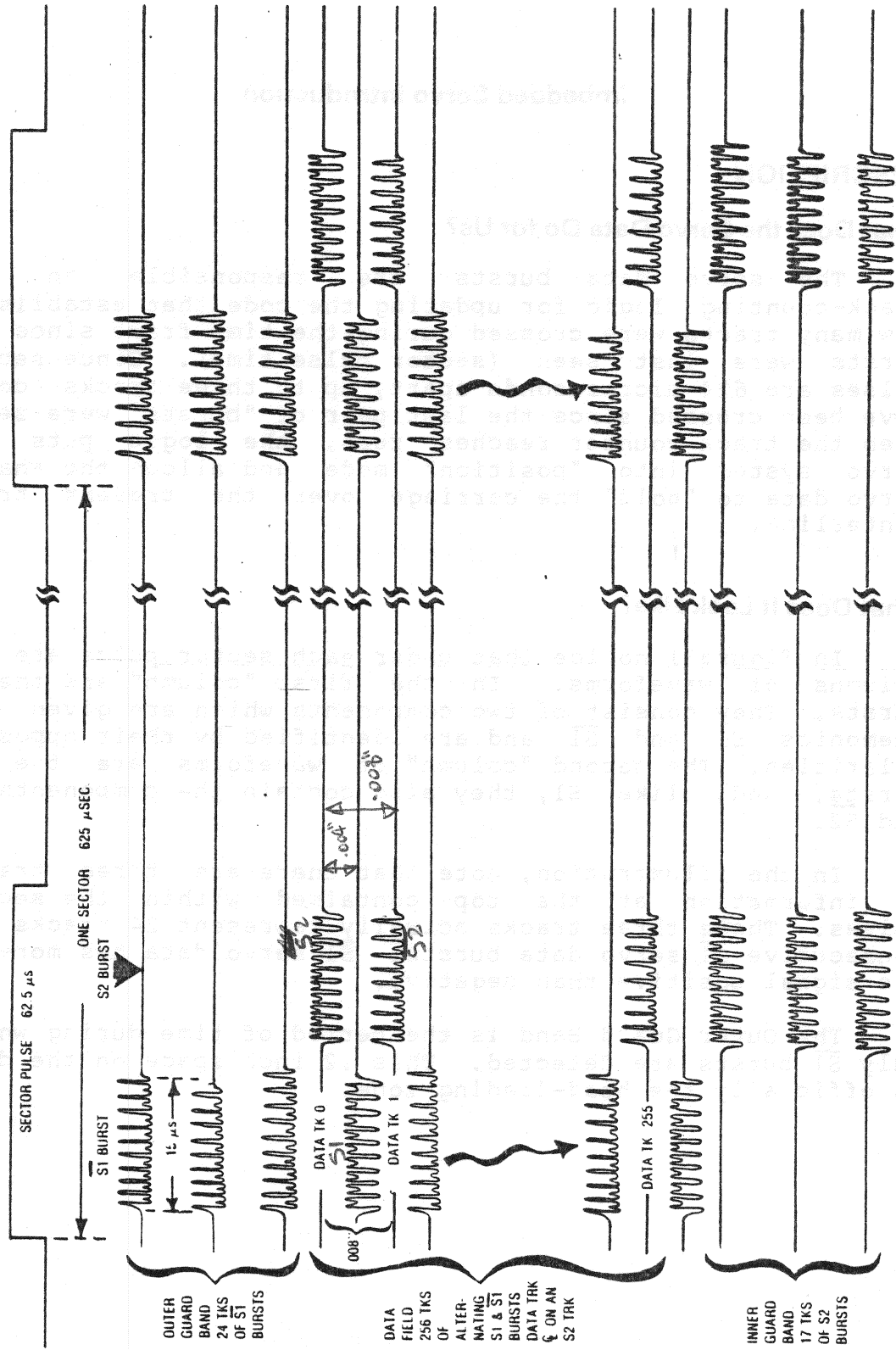


Figure 1. RL01 Servo Data Format

Following the head-loading zone is an area on each surface consisting of alternating  $\overline{S1}$  and S1 bursts. The alternating of these bursts comprises the 256 track data area of the disk surface ( $\approx 2''$ ). Also found in this data area are the S2/ $\overline{S2}$  bursts which are displaced in time and written between the S1/ $\overline{S1}$  bursts. S2 bursts start in the data area and continue into the Inner Guard Band area (where S1/ $\overline{S1}$  bursts are non-existent).

### How Are These Servo Bursts Handled?

Figure 1 is drawn to represent what the disk formatter at Mountain View, Ca. writes. The following examples illustrate how the R/W heads handle this data: if a Read head is centered directly over an  $\overline{S1}$  burst, the head waveform is as shown in Figure 2.



Figure 2.  $\overline{S1}$  Bursts

If the carriage moves over an S1 burst, the Read signal is:



Figure 3. S1 Bursts

Now examine this example of the R/W head centered directly between the the  $\bar{S}1$  and  $S1$  bursts:

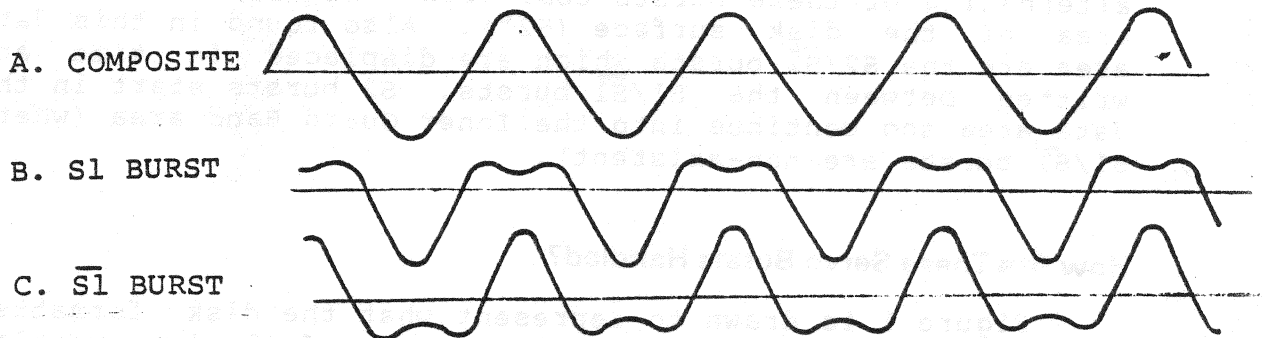


Figure 4. Composite Burst

By looking carefully at B and C, you should be able to visualize how the sine wave occurs. As the heads move off the track centerline slightly, the sine wave distorts. Keeping the R/W heads centered between them creates a perfect (50% duty cycle) sine wave.

Taking all of this one step further, let us examine what happens to these analog signals as they pass through the Read circuits. Built on-board the R/W module are cross-over detectors that sense when the Read signal reaches a zero-volt threshold. Below is an example of an  $\bar{S}1$  servo burst passing through the cross-over detectors:

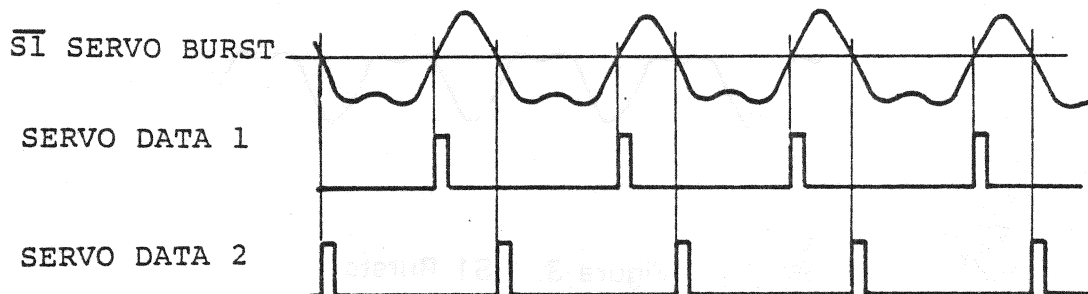


Figure 5.  $\bar{S}1$  Burst Passing Through Cross-Over Detectors

The servo data 1 and 2 pulse trains now become the output of the R/W module and act as source information for the integrator circuits on the drive logic module. These pulse trains then go through one more change to make them into one composite pulse train. See Figure 6.

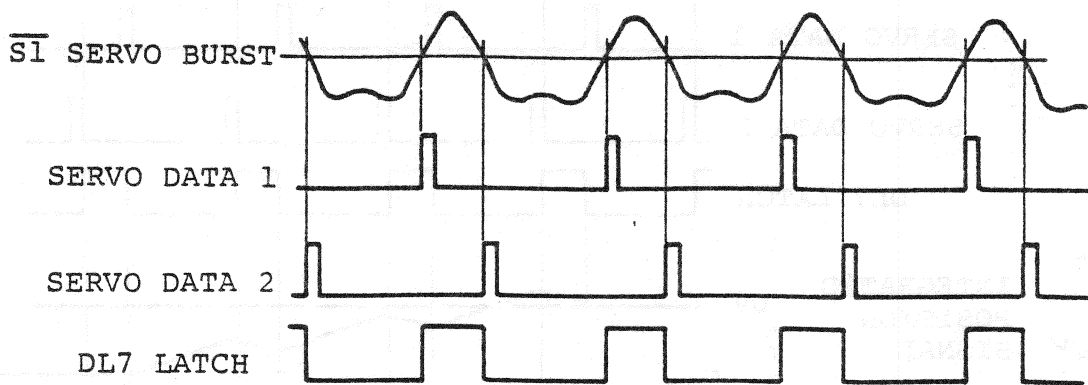


Figure 6. DL7 Latch

The servo data 1 and 2 pulse trains alternately set and clear a latch, producing a composite pulse train which is called the DL7 latch.

FOR INTERNAL USE ONLY

Notice that on Figure 6 the DL7 pulse train output is asymmetrical. If this servo burst signal is passed through an integrator, the result is that the charging network charges more in the negative direction than in the positive. This produces a negative DC voltage, which we will call the position signal. See Figure 7.

*Integrated to keep info. because Servo info goes away.*

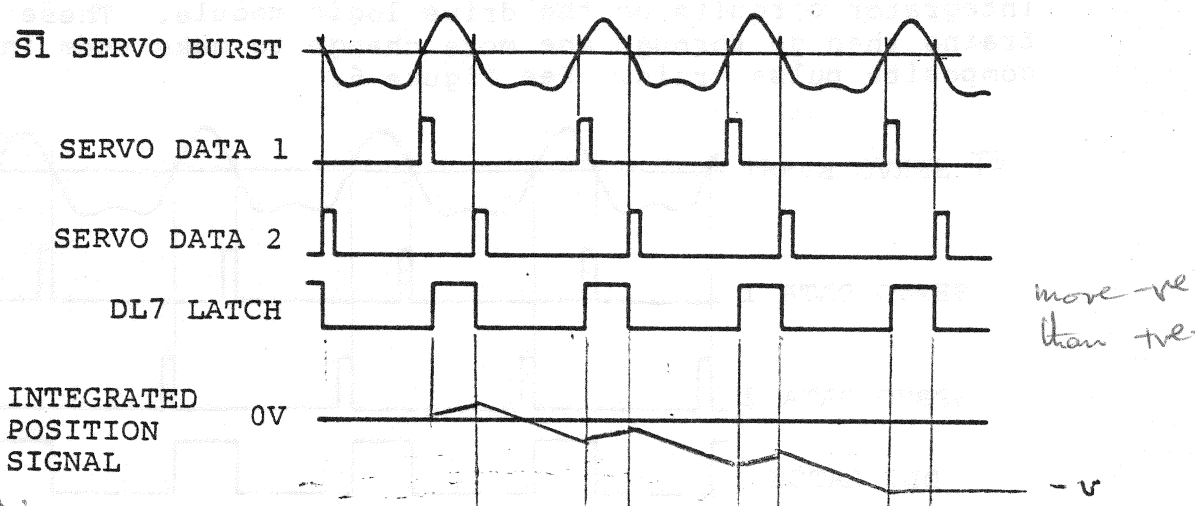


Figure 7. Integration of  $\overline{S1}$

Following is a summary of what you have learned in the last several paragraphs to an examination of the composite signal (Figure 1):

1. An analog waveform is changed into two digital pulse trains.
2. These are changed into a single pulse train.
3. This is changed into a DC voltage level representing the symmetry of the final waveform.

Figure 8 illustrates the generation of the position signal with the R/W heads over a data track centerline. Note that the position signal is a zero-volt signal because the DL7 latch waveform is perfectly symmetrical. The charging network in the integrator circuit will charge the same amount of positive as it does negative, thus producing a zero-volt level.

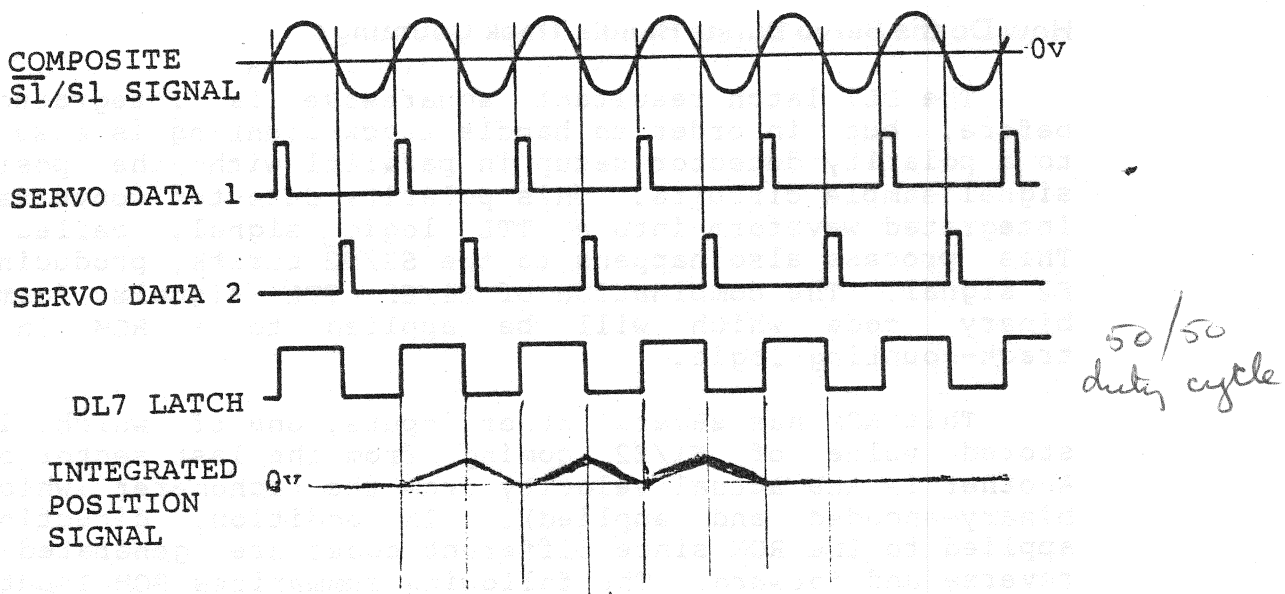


Figure 8. Integration of  $S1/\overline{S1}$  Composite

## How Do the Servo Bursts Handle Track Counting?

The DL7 latch resultant squarewave is integrated as before, but in order to handle track-counting is also sent to a polarity detector setup in parallel with the position signal sample circuits. This polarity detector converts the integrated waveform into a TTL logic signal, called E1. This process also happens to the S2/S $\bar{2}$  bursts, producing an E2 signal. The combination of E1/E2 TTL signals forms a binary code which will be applied to a ROM in the track-counting logic.

This ROM has several other inputs, one of which is a stored value of E1/E2 coming from the last sector seen. Another is the actual velocity from the tachometer (which is binary-encoded and applied). In addition, direction is applied to the ROM since different codes are generated for reverse and forward. The following summarizes ROM Inputs:

1. current E1/E2 samples
2. past values of E1/E2 samples
3. actual carriage velocity
4. direction

These four inputs will allow the ROM to decide how many count pulses will be applied to the track difference counter to decrement it. With each new track count there is a new velocity, and with a new velocity a new input to this ROM. The track difference counter is in turn decremented, the number of times depending upon the number of tracks actually crossed.

cont

This page is for notes.

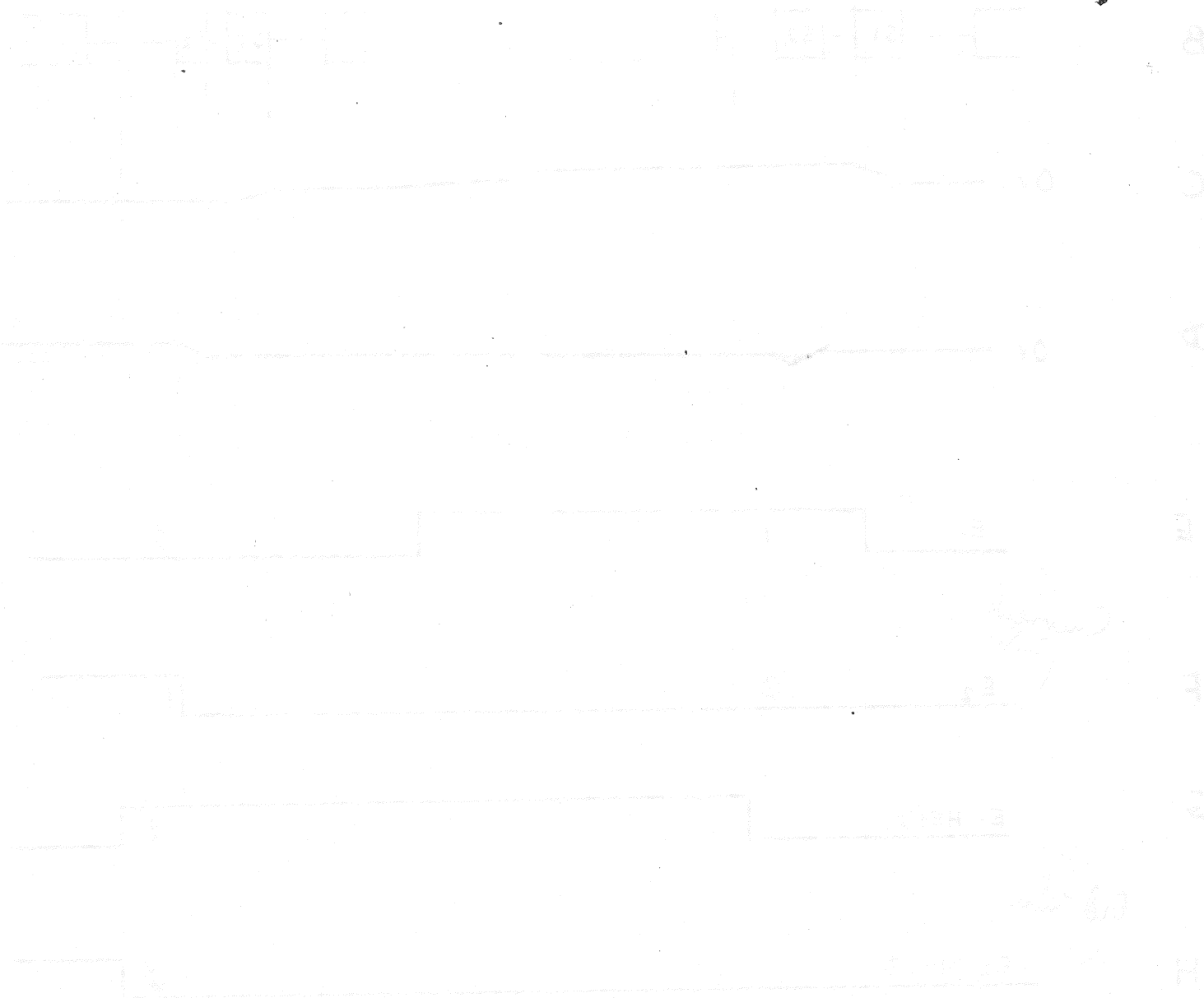
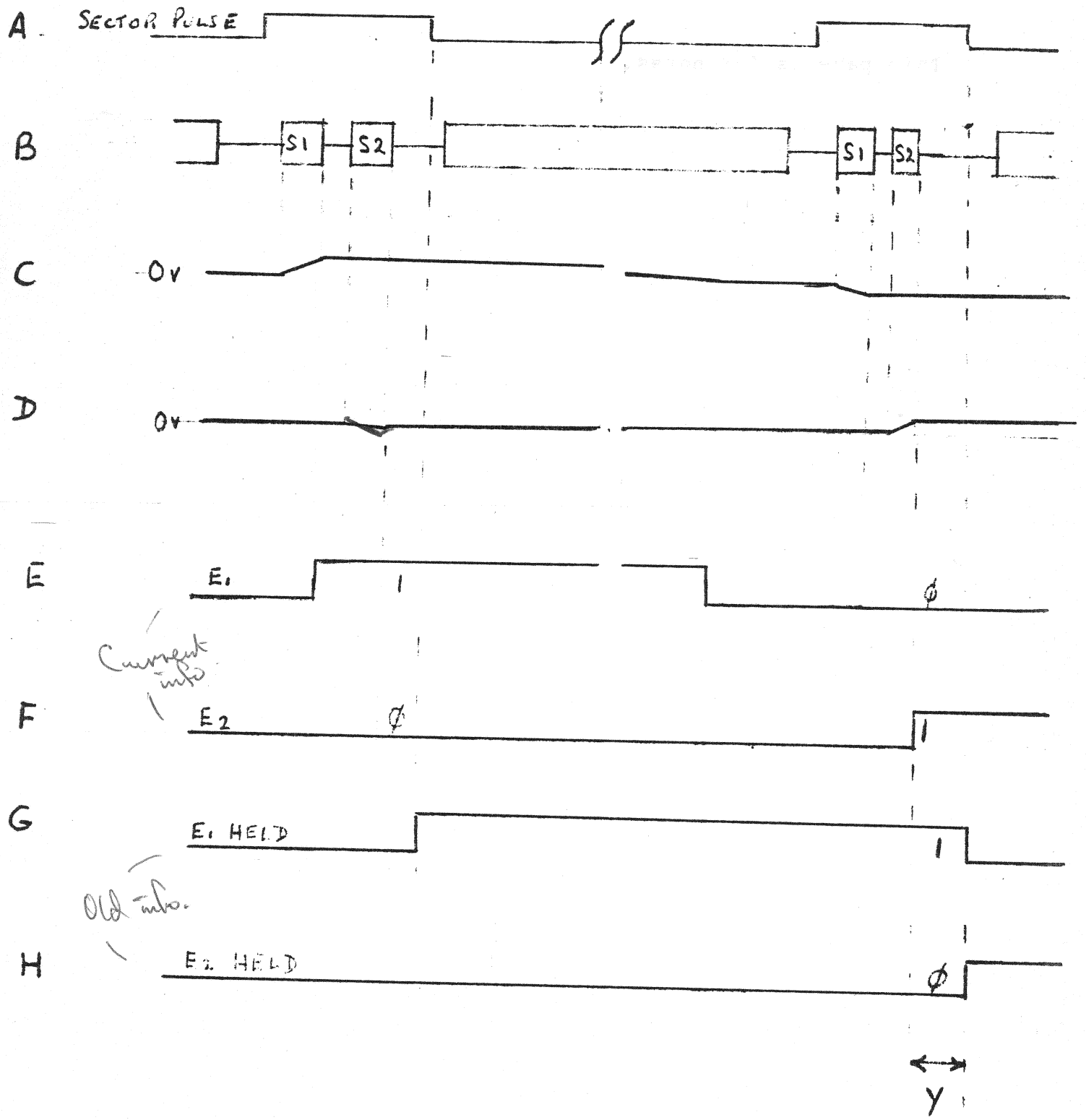


Fig 9.



Let us look at track counting in more detail.

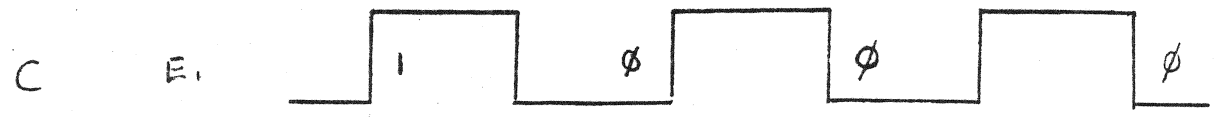
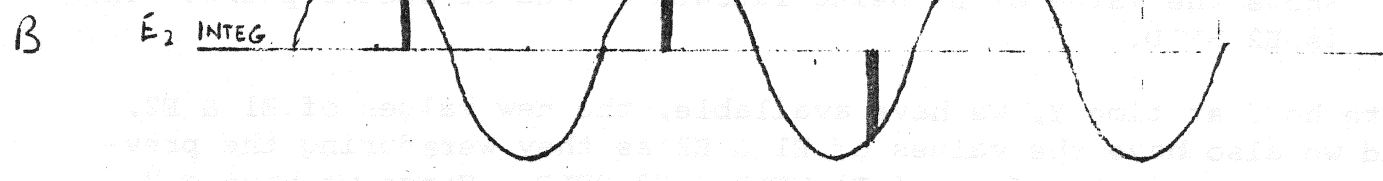
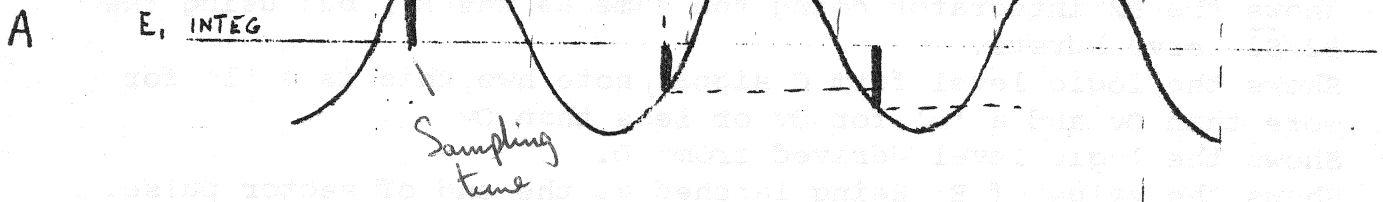
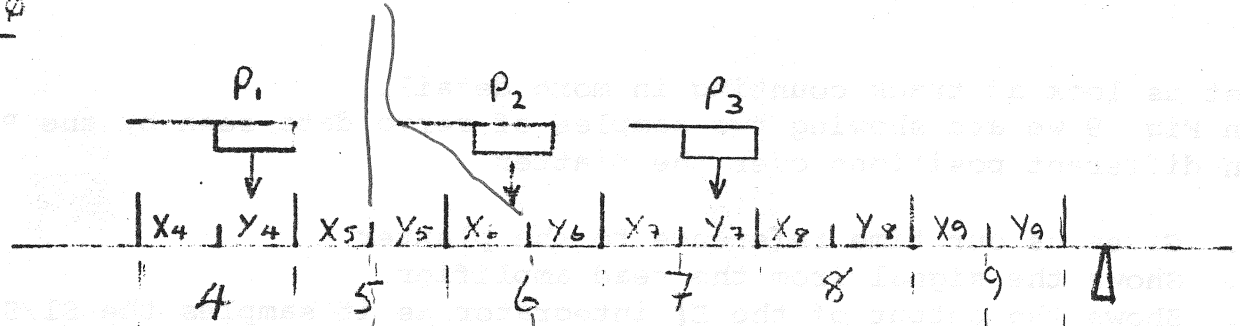
In Fig. 9 we are showing two samples of servo data seen by the R/W Head in different positions over the platter.

- A. Gives us our time reference to the platter
- B. Shows the signal from the read amplifier
- C. Shows the output of the  $E_1$  integrator as it samples the  $S1/S\bar{1}$  burst of servo, note how, during the first sample, the servo has caused a positive voltage which then decays to 0v during the sector, and how, during the second sample the servo has produced a negative voltage, which again, will decay during the next sector.
- D. Shows the  $E_2$  integrator doing the same as the  $E_1$ , but using the  $S2/S\bar{2}$  servo bursts.
- E. Shows the logic level form C signal, note how, this is a '1' for more than 0v and a '0' for 0v or less than 0v.
- F. Shows the logic level derived from D.
- G. Shows the value of  $E_1$  being latched at the end of sector pulse. This is known as  $E_1$  HELD.
- H. Shows the value of  $E_2$  being latched at end of sector pulse. This is  $E_2$  HELD.

Note how, at time Y, we have available, the new values of  $E_1$  &  $E_2$ , and we also have the values of  $E_1$  &  $E_2$  as they were during the previous sample, in the form of  $E_1$  HELD &  $E_2$  HELD. Hence we have a 2 bit code of the OLD and NEW positions of the R/W head at the sample times which of course occur every 625  $\mu$ s (Sector pulse time).

fig 10

centre



$\phi$  volts out of  $E_1$  &  $E_2$  integrators when on track centre.

Let us now consider these samples relative to position of the R/W head over the platter.

In figure 10 we have shown a head in 3 positions ( $p_1, p_2$  &  $p_3$ ), over the platter, the surface of which is divided into the boundaries of cylinders. Each cylinder is further divided into 2 parts X & Y.

- A. Shows the O/P voltage of the E<sub>1</sub> integrator for every position on the disk if a sample occurred at that time.  
Note how, since E<sub>1</sub> monitors the S1/S1 servo, that the value is 0v when on cyl centre. Also that the slope is negative for an ODD Cyl and positive for an EVEN cyl.
- B. Shows the O/P of the E<sub>2</sub> integrator which is 0v when the R/W head is half way between cylinder centres.  
E<sub>1</sub> is the TTL level and is a '1' when  $A > 0v$  and a  $\emptyset$  when  $A < 0v$ .  
E<sub>2</sub> is the TTL level of B.

Now consider the values of E<sub>1</sub> & E<sub>2</sub> when the head is at position P<sub>1</sub> and a sample of servo the processed.

We can see that the E<sub>1</sub> value is '1' and E<sub>2</sub> is ' $\emptyset$ '.

Let us now assume that when the next time a sample of servo is processed (625  $\mu$ s later) the head in position P<sub>2</sub> then:-

New values of E<sub>1</sub> & E<sub>2</sub> are  $\emptyset$  and 1 respectively. Hence the track count ROM which is looking at E<sub>1</sub>, E<sub>2</sub>, E<sub>1</sub> HELD & E<sub>2</sub> HELD would say we have crossed two CYL Boundaries, and would produce TWO count pulses to decrement the difference counter.

Let us further presume, that the effect of the new difference value produced a decrease in velocity of the head, such that when the next servo samples are processed the heads are at position P<sub>3</sub>, then the NEW value of E<sub>1</sub>, & E<sub>2</sub> is  $\emptyset$  &  $\emptyset$  whilst the OLD value is  $\emptyset$  & 1 (from P<sub>2</sub>).

This would mean to the ROM that the heads are either in the region of Y<sub>9</sub> or in the position P<sub>3</sub>.

Well, since the ROM has a velocity input, it knows that P<sub>3</sub> is the actual position, so it produces ONE count pulse, since ONE CYL Boundary has been crossed. If we had been exceeding a velocity threshold, then it would produce 3 count pulses.

Now if you care to work out the codes produced with the LReads moving in reverse then you will see why the ROM must also take into account the direction as well as E<sub>1</sub>, E<sub>2</sub>, E<sub>1</sub> HELD, E<sub>2</sub> HELD & Velocity thresholds

## SUMMARY

Figures 2 through 8 illustrated the development of a position signal. This signal is directly responsible for controlling the carriage movement while in "lock-on" or "position" mode of operation. As the carriage tends to move off-track, the position signal moves from its zero-volt reference to some voltage level (either plus or minus). The amount and direction serve as negative feedback to the servo system to restore the carriage back to the centerline. In track counting, a binary code applied to a ROM delivers to the track-counting logic a number of count pulses (up to three) to decrement the counter. The number of pulses varies due to the speed of the carriage, which may possibly cross three tracks of data in between detection of sector pulses (where the servo data is).

Figure 9 illustrated how the servo bursts from the Read Amplifiers are processed to give a binary code of the present analogue position (E1 & E2), and how this code is latched after use, to become the remembered OLD position code (E1 Held & E2 Held), during the next sample time.

Figure 10 illustrated the value of the analogue voltages for any head position on the disk, and the resultant Binary code which would result from a sample taken at any position on the disk.

