N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE

NASA Technical Memorandum 82853

A High Speed Implementation of the Random Decrement Algorithm

Louis J. Kiraly
Lewis Research Center
Cleveland, Ohio

Prepared for the
1982 Aerospace/Test Measurement Symposium
sponsored by the Instrument Society of America
Las Vegas, Nevada, May 2-6, 1982

NASA

A HIGH-SPEED IMPLEMENTATION OF THE RANDOM DECREMENT ALGORITHM

Louis J. Kiraly
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

E-1192

ABSTRACT

A high speed implementation of the random decrement signal processing algorithm for digitized data has been developed at the NASA Lewis Research Center. The random decrement algorithm is useful for measuring net system damping levels in stochastic processes and for the development of equivalent linearized system response models. The algorithm works by summing together all subrecords which occur after a predefined threshold level is crossed. The random decrement signature is normally developed by scanning stored data and adding subrecords together. The high speed implementation of the random decrement algorithm exploits the digital character of sampled data and uses fixed record lengths of 2^n samples to greatly speed up the process. The contributions to the random decrement signature of each data point is calculated only once and in the same sequence as the data were taken. A hardware implementation of the algorithm using random logic is diagrammed and the process is shown to be limited only by the record size and the threshold crossing frequency of the sampled data. With a hardware cycle time of 200 ns and a 1024 point signature, a threshold crossing frequency of 5000 Hertz can be processed and a stably averaged signature presented in real time.

INTRODUCTION

The random decrement algorithm is used to time average long data records to remove random portions of the signal. When studying a particular system subjected to random excitation, the algorithm averages out random noise as well as the system response to random inputs. It leaves only the averaged system response to a simply defined input. For example, the output of a noisy displacement transducer on a cantilever beam can be processed to determine the beams' response to a simple step displacement excitation.

The algorithm does not require any frequency domain transformations and therefore, requires no presumptions of linearity. The only requirements are that the system be randomly excited and have an ergodic response. These requirements are often met approximately in nature and can be readily simulated in testing laboratories.

Vibrating systems can be studied with this algorithm to determine an averaged or 'equivalent' linearized system response to simple inputs, such as step inputs. From these results, system properties can be back-calculated. For example, the processed response of the cantilever beam can be used to determine the ratio of stiffness to damping and the damping factor for the fundamental mode of vibration. If many points on a vibrating system are processed, normalized stiffness and damping matrices can be back-calculated.

The ability to back-calculate system properties from measure data is useful for determining adequate mathematical models of complex dynamic systems. For example, determinations of the equivalent viscous damping of dry friction stick-slip processes can be determined directly from processed data. Changes in the equivalent viscous damping with differing amplitudes can readily determined with the algorithm. Again, there are no prior assumptions on the linearity of the studied system. In fact, changes in the averaged or 'equivalent' linear system response resulting from changes in the processed input level can easily be determined by re-processing saved data. The changes in the 'linearized' response are a direct indication of the nonlinearities of the system. This is one of the advantages of the random decrement algorithm over more conventional frequency domain methods. Frequency domain procedures have implicit linearity assumptions because time domain data is considered to be a set of superimposed spectral components.

Also, there has been some difficulty in making accurate system damping assessments using frequency domain procedures - stemming primarily from the ease of making the required calculations without considering enough statistically significant data. It is difficult to 'read' a random decrement response without a statistically significant number of samples. This, along with the ability to determine equivalent linear system response to a representative input level results in accurate damping assessments.

The algorithm has also been studied as a means of predicting incipient failure in simulated wing spars, for example, by monitoring progressive changes in the random decrement signature of wing

vibrations (1). Further, it has been studied for use in multidimensional system identification (2) procedures. When used to prepare data for time domain modal extraction procedures, it has helped to find and separate closely spaced and low level modal responses from measured data (3). A study of the dynamic characteristics of fixed wing aeroelastic systems (4) contains a detailed discussion of the mathematical basis for the algorithm.

The biggest disadvantages of the algorithm are that it takes many averages to get 'clean' data and with digital processing, relatively long data files of contiguously sampled data need to be maintained.

A high speed implementation of the basic algorithm has been devised which greatly speeds up the calculations. High speed mechanical systems with response frequencies below 2500 Hz can be processed on the fly with a straight-forward hardware implementation of the algorithm. In software, the high speed implementation of the algorithm can greatly decrease the time required to fully develop a sequence of random decrement signatures.

Random Decrement Processing

The random decrement procedure is illustrated in Figure 1. The basic operation involves summing many subrecords of a longer ergodic signal. With appropriate criteria, this summing procedure minimizes random signal components. If the signal represents the response of a randomly excited physical system, the summing procedure will result in a random-dec signature which minimizes the random components of the system response.

Threshold level crossing of the signal is the most often used criteria for identifying subrecords. In Figure 1, the threshold value is identified as Y_S. The signal crosses the threshold at times denoted by t_1, t_2, t_3, etc. These times demark the starting point of each subrecord to be summed. The first two records are identified, time-shifted to start at the same instant and summed at the bottom left of the diagram. Each following subrecord is also summed to this initial result. Notice that both positive and negative crossings of the threshold are used to define subrecords. The bottom right of Figure 1 shows a representative random decrement signature after many sub-records are added together. This curve represents the equivalent linearized system response to a step input of magnitude Y_S (scaled by the number of subrecords added together).

As shown in Figure 2, each subrecord can be thought of as representing a linear system response composed of the superposition of a step response, an impulse response, random response and random noise. The step response can be thought of as occuring due to the prescribed initial displacement; the impulse response due to the initial velocity or slope; and the random response being, of course, random. When many subrecords are added together the random components average to zero. Also, the plus and minus impulse responses average to zero. As shown in Figure 2, this leaves only the system step response in the resulting signature. This result for a single degree of freedom system can be generalized to multidegree of freedom and nonlinear systems as well. For linear systems, the random decrement procedure results in an autocorrelation function. Also, other criteria for subrecords can be used such as monitoring the signal slope for threshold crossings.

A long contiguous record of the signal must often be maintained. With digital methods, relatively long data files are required. It has been difficult to extract and sum the hundreds of subrecords typically required and process data on-the-fly for system frequencies above a few hertz. Experimental requirements at the NASA Lewis Research Center for propulsion system component frequencies as high as 2500 Hz have required a faster implementation of the basic process.

High Speed Algorithm

Figure 1 shows an ergodic signal from which a random decrement signature will be developed. Examination of the figure reveals that any given point in the signal can simultaneously belong to many subrecords. The starting points of the subrecords can be identified as the data are taken. Therefore, the contribution of any given point in the signal to the summation of subrecords should also be calculable as the data are taken. With digital data acquisition, each samples' contribution to its family of subrecords can be calculated and the random decrement sum updated as the data are taken. The high speed algorithm accomplishes this efficiently.

Figure 3 shows an idealized sampled signal. By the time that the thirteenth sample is taken, four subrecords have been identified which start at the zeroth, sixth, tenth, and twelveth samples. The thirteenth sample becomes the thirteenth point in the first record, the seventh point in the second record, the third point in the third record and the first point in the fourth record. So, when calculating the sum of subrecords, point thirteen must effect the first, third, seventh, and twelveth position in the resulting summed records. It would be sufficient to simply add sample thirteen to the already existing sums corresponding to the first, third and seventh positions of the signature. The sample would be simply assigned to signature position thirteen since no prior sum exists there. After a prescribed number of subrecords had been fully processed, each point in the resulting signature would be divided by the number of sub-records used.

With real time processing capabilities, it would also be desirable to watch the signature evolve as data are taken. This is not reasonably possible with the procedure described above. Notice, that by simply adding the thirteenth sample to the first, third, and seventh position, the total number of samples contributing to the averaged record at these points are different. The zeroth and first positions in the resultant record have been summed four times. The second and third position have been summed three times. The fourth through seventh position have been summed twice. With a few more records and samples any real-time display

of data in this format would be incomprehensible. The first several points would have to be divided by the number of records considered, the next few points by one less than the number of records and so forth - so that the overall display could be in the same proportion. To accomodate real time display, stable averaging is used. With stable averaging, the contribution of the next number in a sequence of numbers to the average of the sequence is calculated directly. From Figure 3, the averages at each of the summed record positions can be expressed as:

record 1: $\Sigma_{13} = D_{13}$

record 2: $\Sigma_7 = (D_{13} + D_7) \div 2$

record 3: $\Sigma_3 = (D_{13} + D_9 + D_3) \div 3$

record 4: $\Sigma_1 = (D_{13} + D_{11} + D_7 + D_1) \div 4$

with Σ_j' representing the prior value at location j in the summed record and Σ_j representing the new value, all of the above equations can be replaced simply with,

$$\Sigma_j = \Sigma_j' + (D_i - \Sigma_j') \div n$$

where D_i is the current data sample and n is the record number index. For example,

record 1: $\Sigma_{13} = \Sigma_{13}' + D_{13} - \Sigma_{13}'$

record 2: $\Sigma_7 = (D_7) + (D_{13} - D_7) \div 2$

record 3: $\Sigma_3 = [(D_9 + D_3) \div 2]$
$+ (D_{13} - [(D_9 + D_3) \div 2]) \div 3$

record 4: $\Sigma_1 = [D_{11} + D_7 + D_1) \div 3]$
$+ (D_{13} - [D_{11} + D_7 + D_1) \div 3]) \div 4$

With stable averaging, the summed data can be readily displayed in real-time. The resultant display is in proper proportion and is smoother at the beginning and more ragged looking near the end of the sum record until the required number of subrecords are summed together. This is because the first points of the signature will initially benefit with more averages than the latter points.

The high speed algorithm without stable averaging is shown in Figure 4. Five counters and two blocks of memory are required. Subrecord lengths and the resulting signature size are finite and set at powers of two. Most of the required counters can function using only the least significant bits which can wrap-around many times. The algorithm is greatly sped up since there is very little limit checking required during the assignment of data to the summed random decrement signature. Many of the functions can be done in parallel in hardware systems to further speed the process.

The algorithm has three basic sequences. The first is the INITIALIZE sequence which sets the counters to zero and scans the sampled data continously until a new sub-record is found. The next sequence, ADD A RECORD, updates a table in memory which is associated with the defined sub-records. The final sequence gathers subsequent samples and computes their contribution to the random decrement signature for each sub-record of which the sample is part.

The SHIFT counter in Figure 4 is incremented every time a sample is taken. Each position in the table memory is associated with an identified sub-record. When a new sub-record is started because a signal threshold has been crossed, the negative of the current SHIFT count is stored in the table position which corresponds to the new sub-record. This position corresponds to the current STOP count.

The current sub-record under consideration is pointed to by COUNT. When subsequent data are taken, the updated SHIFT count is added to the table value to determine which point in the summed signature is to be updated by the sub-record and the new data (i.e., LOC = TBL(COUNT) + SHIFT). The principle is illustrated by examining Figure 3. When the thirteenth data point is taken, the SHIFT counter would also be at thirteen and previously stored table values for the four active sub-records would be 0, -6, -10, and -12. Adding these values to the current SHIFT count points to the appropriate positions in the summing memory. Once a sample is taken, its contribution to the summed signature is calculated for every active sub-record in the table before the next sample is taken (i.e., SUM(LOC) = SUM(LOC) + SAMPLE).

Sub-records and the summed signature are a finite length. When the sampled data goes beyond the length of a sub-record, that sub-record no longer needs to be considered. The START counter points to the first active subrecord in the table memory and the STOP counter points to one position post the last active record. A table counter called COUNT is used to go from the START table position to the STOP position as a sample is being processed. The START counter is incremented when the START table value added to the SHIFT count is greater than the record size (i.e., LOC > MAX). Again, with record lengths chosen as powers of two, this condition is readily checked when the bit corresponding to the record size goes to one. This is easily checked in hardware and rapidly done in assembly coded software. The STOP counter is incremented whenever a new record is started. This happens whenever the sampled data crosses the threshold and a preselected number of records haven't already been started (i.e., RECNO $\leq$ LIMIT).

If the number of table entries are the same as the size of the signature memory, all counters except the record number counter can operate in wrap-around fashion. There can never be more sub-records than there are points in a subrecord so the START, STOP, and SHIFT counters can wrap-around many times without difficulty or any need for checking.

A new sub-record is defined whenever the next data minus the selected threshold changes sign from the last data sample. Since the data are sampled, neither the prior sample nor the current sample

are likely to be zero exactly (i.e., on the threshold). It is desirable to consider the sample closest to the threshold as the starting point for the new record. As discussed earlier, the apparent starting point of the new record is controlled by storing the negative of the SHIFT count in the table. If the current sample is closest to the threshold (i.e., |OLD| < |NEW|), the current SHIFT count is used. If the prior sample is closest to the threshold, the stored value is one less. This, in effect, shifts the record location by one position. The current data sample is presumed to be exactly on the threshold for the first location of the new sub-record, and results in as much as a one-half sample period time error.

Stable averaging can be added to the algorithm during the assignment of data to the signature memory as shown in Figure 5. Figure 7 shows the process schematically for a hardware implementation. An absolute count of the number of records (RECNO) is required which can be decremented (RECDEC) as each active sub-record is scanned. It is unlikely that stable averaging would be used in strictly software versions of the algorithm. Stable averaging and the extra counters can be added in parallel to hardware implementations with minimal impact on throughput.

Hardware Implementation

Figure 6 is a block diagram for a hardware implementation of the algorithm. Data is clocked out of an analog to digital convertor with a sample rate clock into a first-in-first-out (FIFO) buffer. The FIFO buffers the extra cycles that the system adds when a new record is started. The FIFO allows data to be sampled at the maximum possible average rate. The data entry logic is used to subtract preselected threshold values from the input data and to signal when a new sub-record is formed. The counter save and address logic maintains the wrap around START, STOP, SHIFT, and table pointing counters. Also, this logic maintains the table and signals various conditions. Stable averaging logic combines the current sample with saved data so that a stably averaged signature is maintained in memory. The processing and display memories and associated logic prepare and maintain the signature for presentation.

The addressing scheme is shown in Figure 7. All of the active record positions in the table memory are read in sequence. The stored values in the table are added to the current shift count value. This new value points to the particular position in the processing memory that needs to be updated by the current sample. The value from this position is stably averaged with the current sample and the result is stored in the processing memory as well as in the display memory. The processing memory is read somewhat randomly in accordance with the computed index. The display memory, however, is read out in sequence. Between display memory read cycles, updated signature values are written simultaneously into the display and processing memories.

Figure 8 is a schematic of the data entry logic. Wherever standard transistor-transistor logic parts are used, their part number is given in parenthesis at the upper right. The resets and clocking logic are shown in Figures 11 and 12. A fourteen bit data word is sampled, a selected threshold is subtracted, and the resulting sixteen bit word is designated XNEW. $\overline{\text{SGNCHG}}$ is generated whenever the subtraction results in a sign change from the last sample. The relative absolute value of the last sample compared with the new sample is signaled by SELXO. This signal is used when a new record is started to shift the apparent starting point one position, if needed. The status of the data collection system is flagged by the signals $\overline{\text{FIRDY}}$ and FIFULL which denote that a new sample is available, or that data are being sampled, at too high a rate respectively.

The four counters in the counter save and address logic (Fig. 9) are updated according to conditions previously discussed. A representative timing analysis is given in Figure 13 which shows the relative timing of the signals. The START and TABLE counters are eight bit. The STOP counter is twelve bits so that the absolute number of subrecords (STPCNT) is available for the stable averaging logic. The shift counter is ten bits to address a 1024 word long signature memory.

All of the counters as well as the least significant eight bits of the STOP counter are allowed to wraparound. After 4096 subrecords have been found, LIMIT is generated to terminate the process irregardless of the selected number of records.

STOP is generated whenever the START counter and the STOP counters' least significant bits are equal. They are equal only when there are no active subrecords to process or when the average rate of new-subrecord creation is greater than one every four samples. The latter condition would result in a signature and signature display so poor that it is highly unlikely that the system would be operated in that manner.

PASS is generated whenever the last active record in the table memory has been processed. PASS and the status of the FIFO are used to bring the next data sample into the system.

NEWRC is generated when a new threshold crossing occurs and the pre-selected number of subrecords have not yet been found. The STOP counter points to the next available position in the TABLE MEMORY and the negative of the SHIFT counter is stored at that position on the low transition of $\overline{\text{WEI}}$.

The processing memory address to be updated is generated in the SUM ADDRESS adder. If this address is greater than the processing memory length (1024 positions) an OVFL signal is generated which updates the START counter.

The processing and display memory with stable averaging is shown in Figure 10. The division required by stable averaging is done by multiplying by a PROM generated reciprocal. The DATA-SUM adder, SUM LATCH, MULTIPLYER, AND UPDATE ADDER do the stable averaging. If the first subrecord is still active, the RECORD DOWN COUNTER will get to one. Then, $\overline{\text{ZSUM}}$ is generated and any previous

signature summation data will be ignored at the SUM LATCH. When the number of the sub-record being processed exceeds 1024, the reciprocal generated for the multiplyer becomes vanishingly small and SX0 sets the DATA-SUM value to zero; as if it were divided by a large digital number. MORRECS is generated by a comparator to indicate that there are still more subrecords desired to be found.

The display counter clocks the display memory in sequence through a digital to-analog convertor and generates a trigger for an oscilloscope display.

Figure 11 shows the master 20 MHz clock used to drive the system and the conditional clocking functions. Figure 12 shows other required logic and some status indicators. The first sequence in figure 13 shows representative timing for table entries being processed and a new subrecord being added. The second sequence shows table entry processing without a new subrecord being added. All of the timing can be derived from the system schematics.

Hardwired Algorithm Performance

The maximum throughput is limited by the rate of which new subrecords are added for processing. With N samples per record and an analog-to-digital conversion (ADC) rate of r samples/second, the time spanned by a single record t_r, becomes

$$t_r = N/r$$

With R as the average threshold crossing frequency, the average number of active records that a sample can belong to is

$$M = Rt_r = RN/r$$

with t_m representing the time required to read memory, process the values and store data back into memory again, the total time T required to process a sampled data point is:

$$T = Mt_m = RNt_m/r$$

For continuous throughput, the time T must be less than the time required per sample:

$$1/r > T \quad \text{or} \quad 1/r > RNt_m/r$$

and

$$R < 1/(Nt_m)$$

This expression indicates that the maximum threshold crossing rate is a function of the number of samples in the signature and the processing time per sample only, and is independent of the ADC rate. This means that a given signal can readily be time-scaled for display purposes without regard for algorithm performance.

For the system shown in Figures 8 through 13, the maximum threshold crossing rate is 5000 Hz. This corresponds to a maximum frequency of 2500 Hz in the processed signal.

SUMMARY AND CONCLUSION

A high speed implementation of the random decrement algorithm for sampled data has been presented which allows each sample to be handled only once and as it is taken. Each samples' its contribution to a random decrement signature is calculated as it is taken. With the algorithm, software calculations using assembly code can be made very quickly. Also, hardware versions of the algorithm can be made to execute very fast and even process high frequency data on the fly.

The principle of stable averaging was discussed for use in real time displays and a hardware schematic was developed.

A hardwired version of the algorithm was developed and the timing analyzed. Performance analyses showed that a straight forward design with standard transistor-transistor-level logic and a 1024 word long signature could process data with frequencies as high as 2500 Hz in real time. Shorter length signatures and faster logic can speed the process even more.

A hardwired system based on the schematics presented is under development and will be the subject of a future paper.

REFERENCES

(1) Coe, H. A. Jr., "On-Line Failure Detection and Damping Measurement of Aerospace Structures by Random Decrement Signatures," NASA CR 2205, March 1973.

(2) Ibrahm, S. R., "Random Decrement Techniques for Modal Identification of Structures," J. Spacecraft, Vol. 14, No. 11, Nov. 1977.

(3) Ibrahm, S. R., and Mikulcik, E. C., "The Experimental Determinations of Vibration Parameters from Time Responses," Shock and Vibration Bulletin, Bull. 46, Pt. 5, August 1976.

(4) Chang, C. S., "Study of Dynamic Characteristics of Aeroelastic Systems Utilizing Randomdec Signatures," NASA CR 132563, 1975.

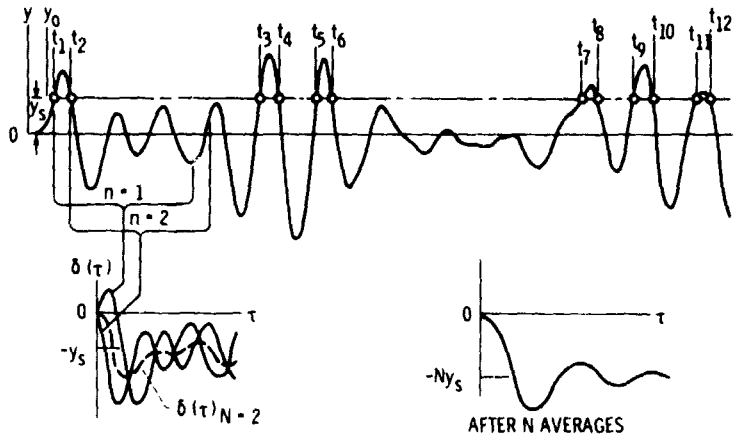

Figure 1. - Random decrement process.

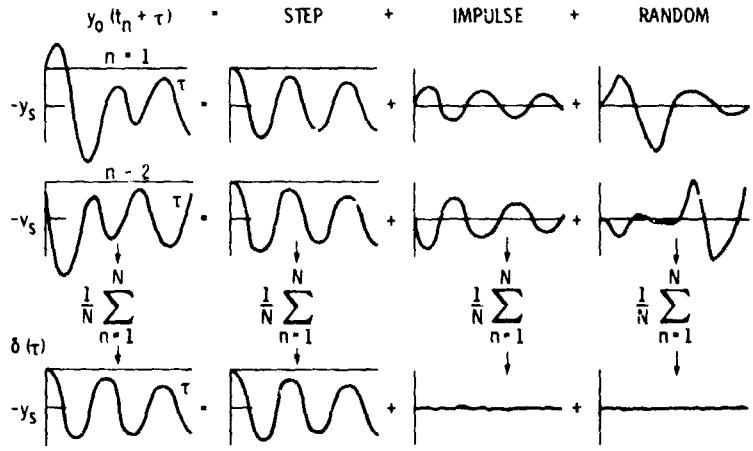

Figure 2. - Removal of impulse and random components with averaging.

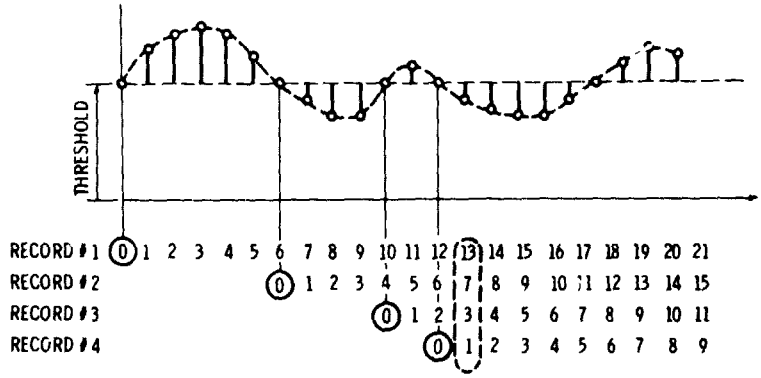

Figure 3. - Random decrement with digitized data.

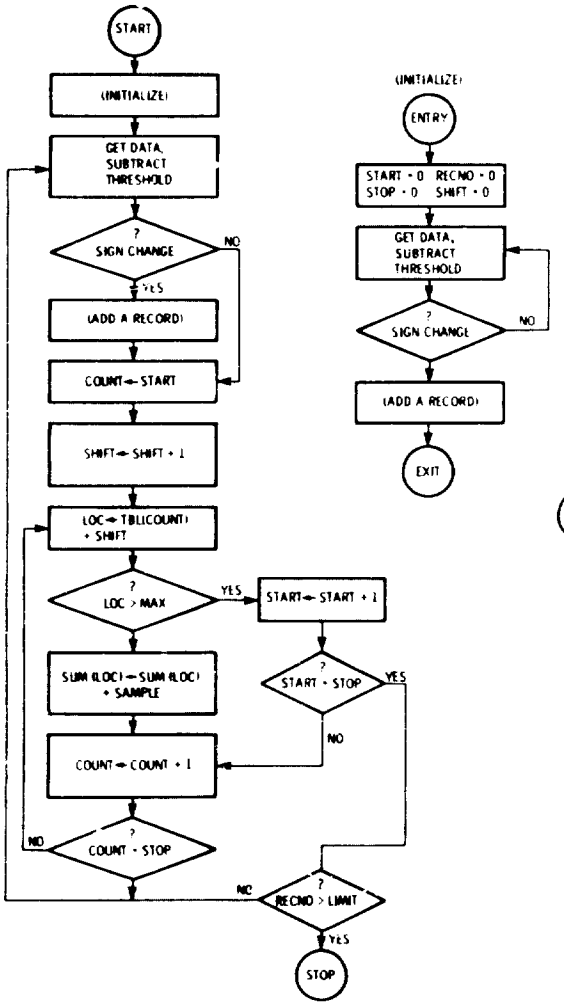

Figure 4. - High speed algorithm without stable averaging.

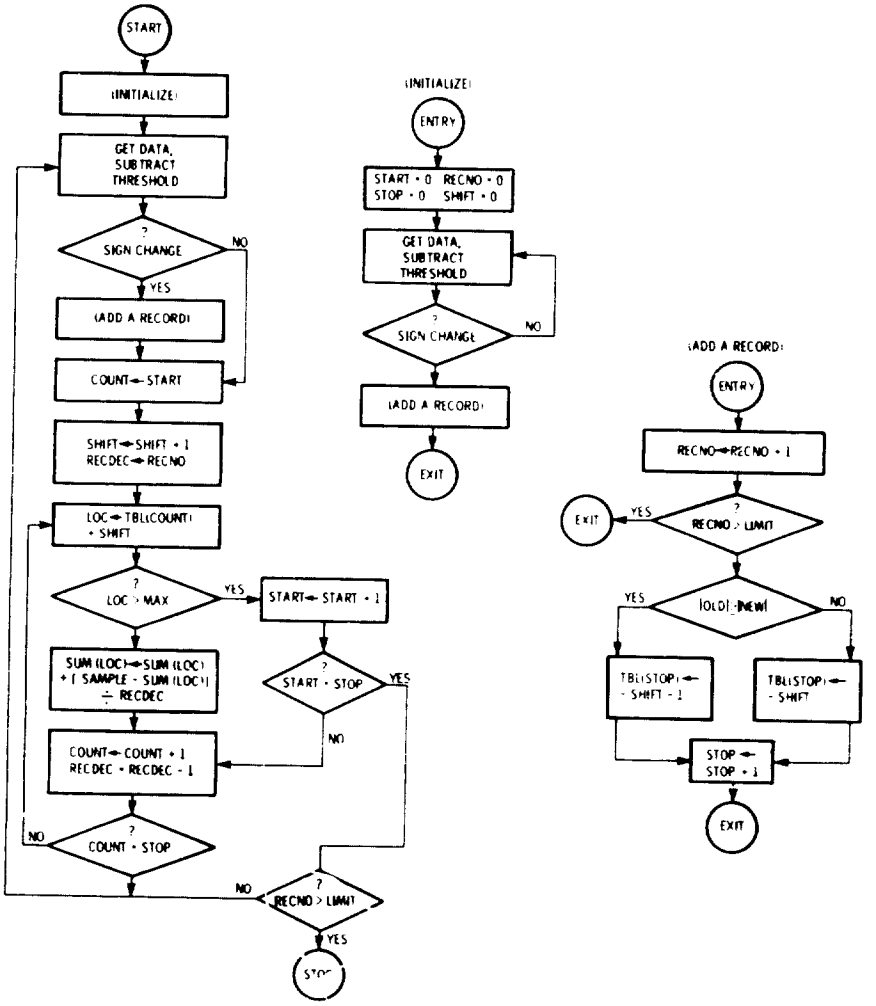

Figure 5. - High speed algorithm with stable averaging.

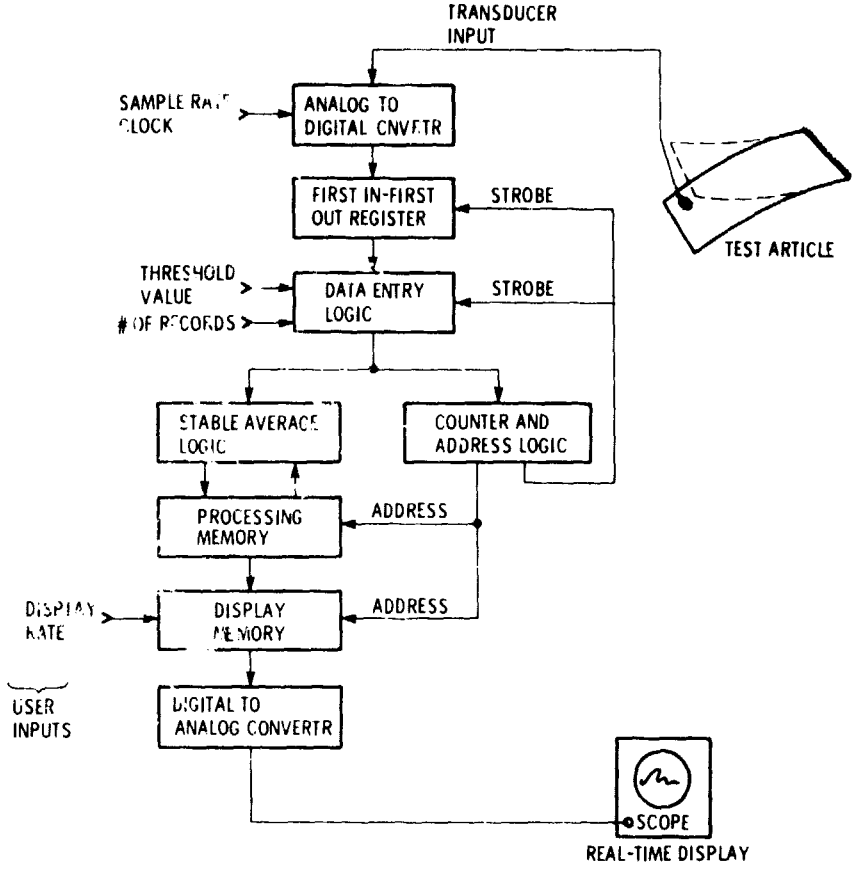

Figure 6. - Block diagram for hardware implementation of high speed algorithm.

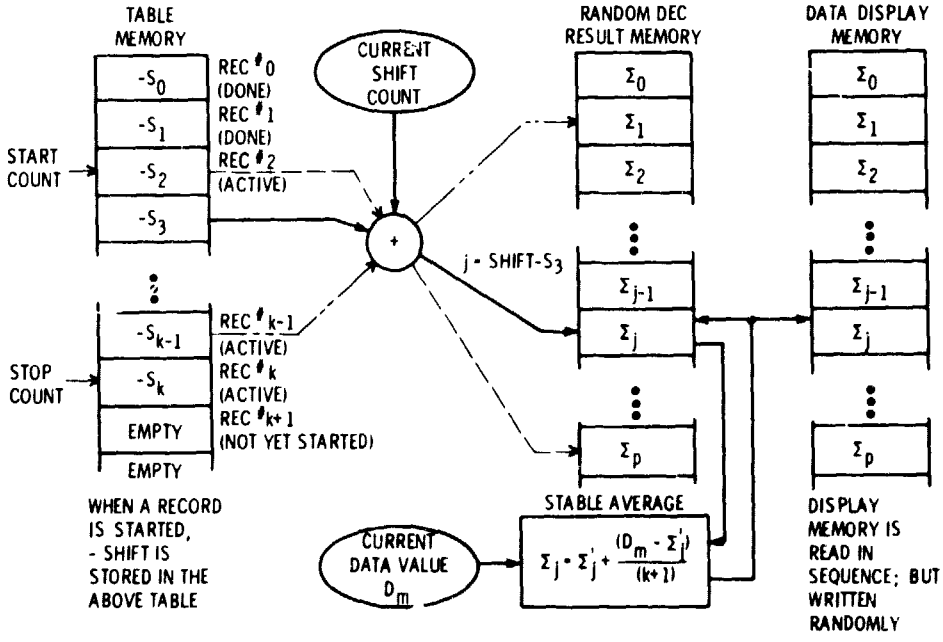

Figure 7. Memory access schematic.

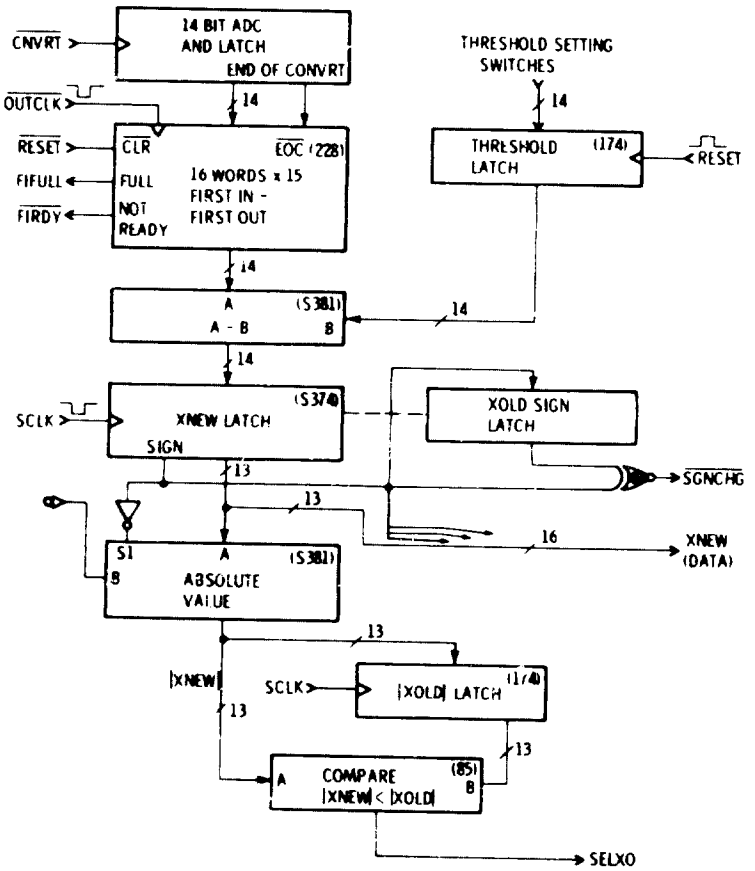

Figure 8. - Data entry logic.

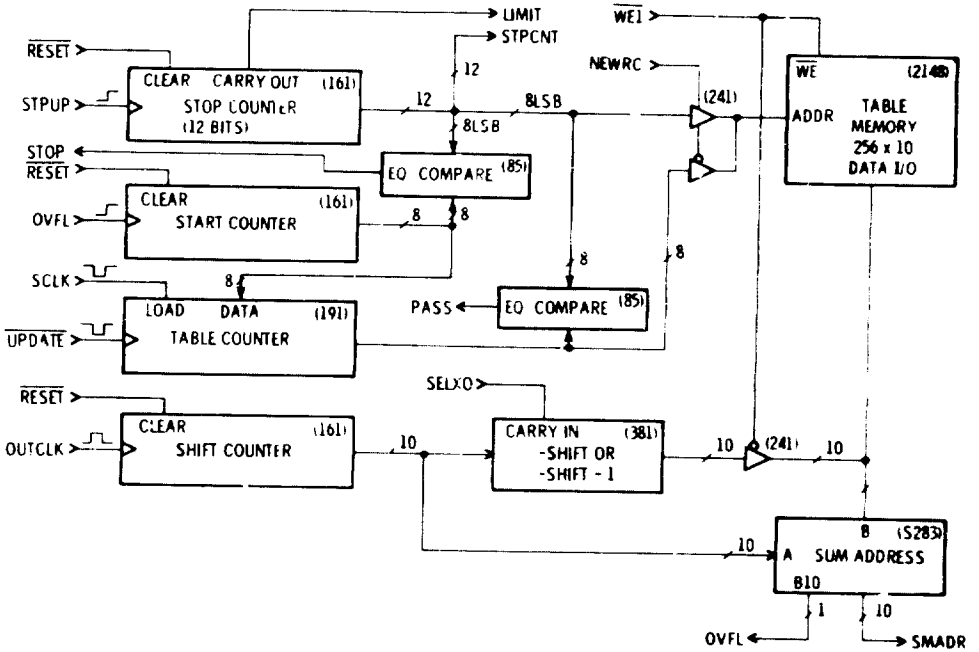

Figure 9. - Counter save and address logic.

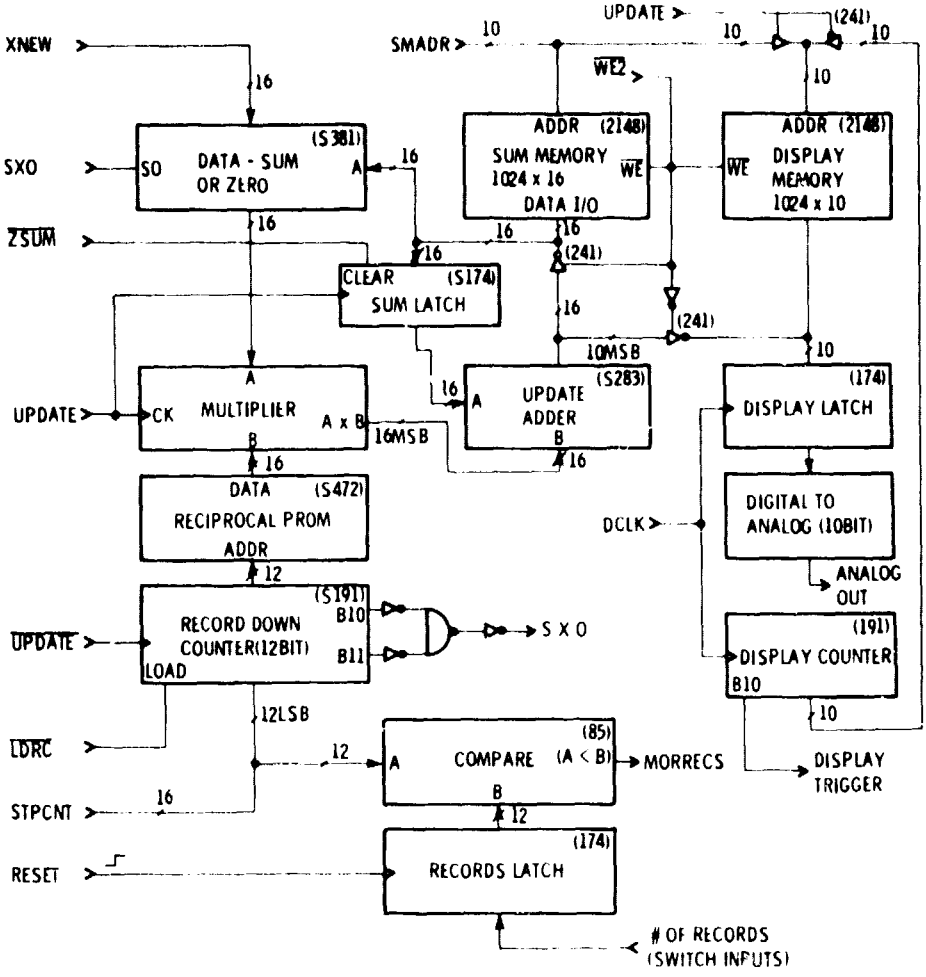

Figure 10. - Processing and display memory.

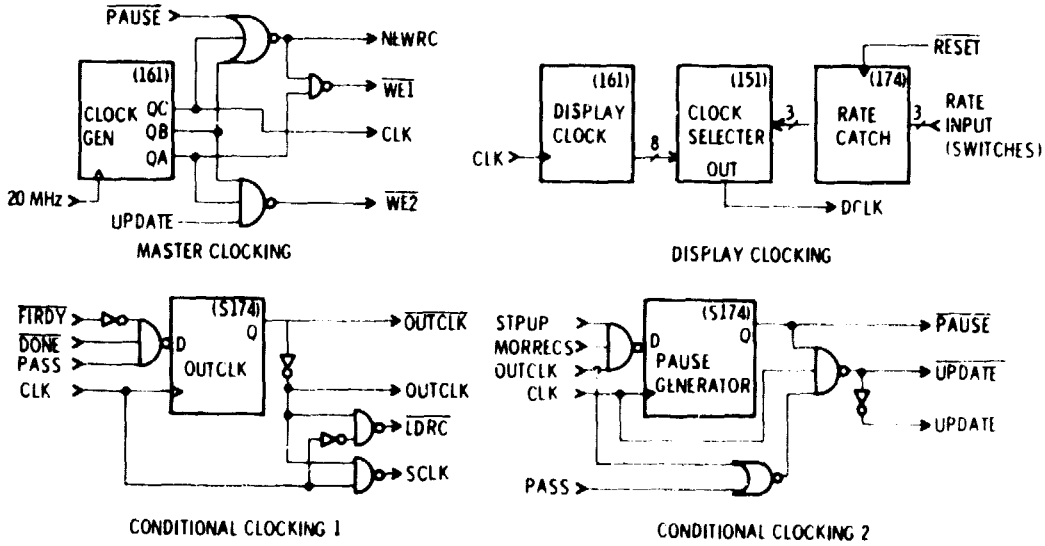

Figure 11. - Clocking functions.

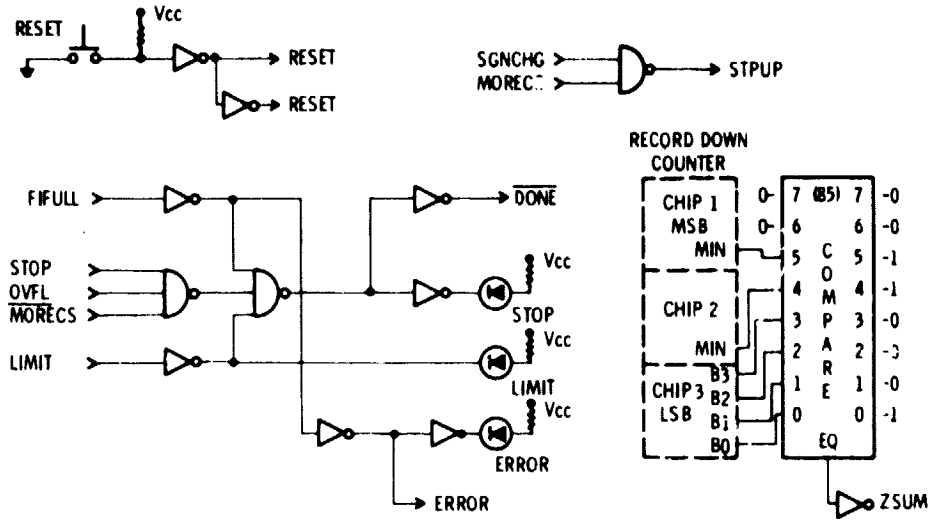

Figure 12. - Miscellaneous logic.

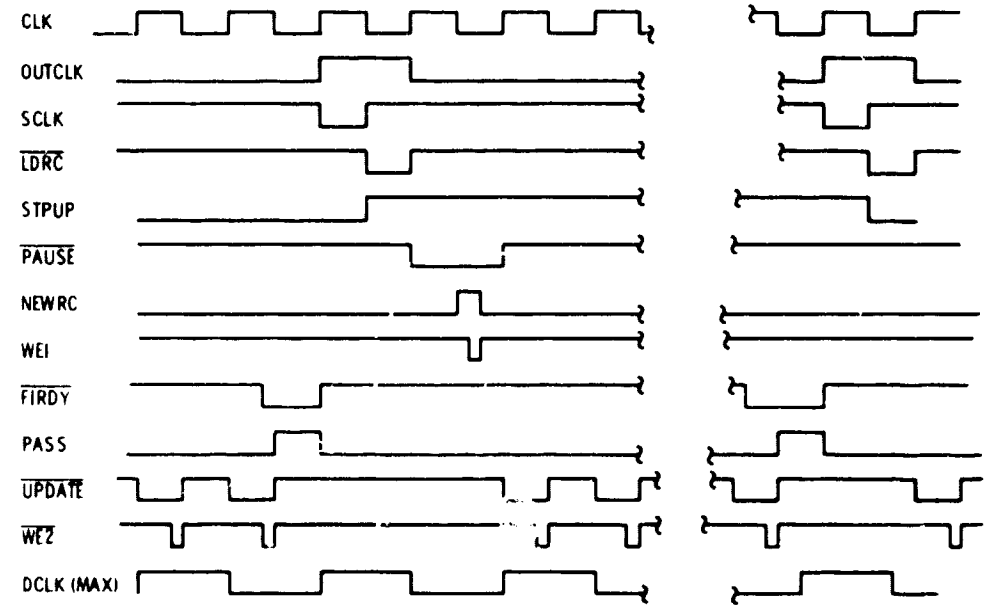

Figure 13. - Representative timing.